

Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky

**Pokročilé metódy riadenia robotického
ramena s integráciou počítačového
videnia**

Diplomová práca

2023

Bc. Oliver Kudzia

Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky

**Pokročilé metódy riadenia robotického
ramena s integráciou počítačového
videnia**

Diplomová práca

Študijný program: Inteligentné systémy
Študijný odbor: Informatika
Školiace pracovisko: Katedra kybernetiky a umelej inteligencie (KKUI)
Školiteľ: doc. Ing. Peter Papcun, PhD.

Košice 2023

Bc. Oliver Kudzia

Abstrakt v SJ

Cieľom tejto diplomovej práce je namodelovať, fyzicky skonštruovať a naprogramovať šesťosé robotické rameno schopné uchopovať objekty na základe vizuálnej odozvy zo systému počítačového videnia s využitím senzoru Kinect. Práca sa zameriava na konštrukčné aspekty ramena a stojanu na kameru, popisuje softvérovú časť riadenia ramena a rozpoznávania objektov v hĺbkovej mape, ako aj architektúru integrácie počítačového videnia do systému riadenia ramena. Súčasťou práce je aj vytvorenie vizualizácie a grafického používateľského rozhrania pre robota. V závere sa prezentujú dosiahnuté výsledky na základe vykonaných experimentov.

Kľúčové slová

robotické rameno, kinematika, riadenie, počítačové videnie, Kinect, Python, C/C++, C#

Abstrakt v AJ

The aim of this diploma thesis is to model, physically construct and program a six-axis robotic arm capable of grasping objects based on visual feedback from a computer vision system using a Kinect sensor. The work focuses on the construction aspects of the arm and stand for the camera, describes the software part of the arm control and object recognition in depth map as well as the architecture of computer vision integration into the arm's control system. Thesis also includes making visualization and a graphical user interface for the robot. Finally, the achieved results based on experiments are presented.

Kľúčové slová v AJ

robotic arm, kinematics, control, computer vision, Kinect, Python, C/C++, C#

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY
Katedra kybernetiky a umelej inteligencie

ZADANIE
DIPLOMOVEJ PRÁCE

Študijný odbor: **Informatika**
Študijný program: **Inteligentné systémy**

Názov práce:

Pokročilé metódy riadenia robotického ramena s integráciou počítačového videnia
Advanced control methods of robotic arm with computer vision integration

Študent: **Bc. Oliver Kudzia**
Školiteľ: **doc. Ing. Peter Papcun, PhD.**
Školiace pracovisko: **Katedra kybernetiky a umelej inteligencie**
Konzultant práce:
Pracovisko konzultanta:

Pokyny na vypracovanie diplomovej práce:

1. Analýza kinematiky robotických ramien
- 2.. Analýza rozpoznávania objektov pomocou hĺbkových máp
3. Návrh robotického ramena a jeho riadenia s integráciou počítačového videnia
4. Realizácia robotického ramena a jeho riadenia s integráciou počítačového videnia
5. Vyhodnotenie riadenia robotického ramena s integráciou počítačového videnia
6. Vypracovať dokumentáciu podľa pokynov vedúceho práce (hlavná časť minimálne 60 strán, prílohy – užívateľská a systémová príručka, tlačená forma v nerozoberateľnej väzbe)

Jazyk, v ktorom sa práca vypracuje: slovenský
Termín pre odovzdanie práce: 21.04.2023
Dátum zadania diplomovej práce: 31.10.2022



prof. Ing. Liberios Vokorokos, PhD.
dekan fakulty

Čestné vyhlásenie

Vyhlasujem, že som diplomovú prácu vypracoval samostatne s použitím uvedenej odbornej literatúry.

Košice 21. 4. 2023

.....

Vlastnoručný podpis

Podakovanie

Chcel by som sa podakovať vedúcemu mojej práce doc. Ing. Petrovi Papcunovi, PhD. za cenné rady a ochotu pomôcť počas vypracovania tejto záverečnej práce. Podakovanie patrí aj doc. Ing. Jánovi Jadlovskému, CSc. za poskytnutie materiálnych prostriedkov na konštrukciu stojanu. V neposlednom rade by som sa chcel podakovať aj mojím rodičom za možnosť rozvíjať svoj potenciál. Na záver ďakujem aj môjmu spolužiakovi Samuelovi Titkovi za jeho kreatívne nápady a poradenstvo.

Obsah

Úvod	1
1 Formulácia úlohy	3
2 Analýza kinematiky robotických ramien	5
2.1 Priama a inverzná kinematika	6
2.2 Pohonné jednotky kĺbov	8
3 Analýza rozpoznávania objektov pomocou hĺbkových máp	11
3.1 Microsoft Kinect V2	11
3.2 Hĺbková mapa	13
3.3 Metódy spracovania obrazu	15
3.3.1 Prahovanie a morfologické operácie	16
3.3.2 Detekcia hrán	18
3.3.3 Konvolúcia	20
3.3.4 Houghova transformácia	22
3.3.5 RANSAC	24
3.4 Rozpoznávanie objektov	25
4 Návrh a realizácia robotického ramena a jeho riadenia	27
4.1 Návrh a realizácia konštrukčnej časti robotického ramena	27
4.1.1 Nástroj Autodesk Fusion 360	27
4.1.2 Tvorba konštrukčného návrhu	30
4.1.3 Realizácia konštrukcie ramena	35
4.2 Návrh a realizácia riadiacej jednotky robotického ramena	39
4.2.1 Návrh architektúry riadiacej jednotky ramena	39
4.2.2 Realizácia riadiacej jednotky ramena	43
4.3 Architektúra riadenia	45
4.3.1 Návrh architektúry riadenia	45

4.3.2	Realizácia architektúry riadenia	46
5	Návrh a realizácia integrácie počítačového videnia do riadenia	52
5.1	Stojan na Kinect a popis výrobného procesu	52
5.2	Návrh architektúry integrácie počítačového videnia	54
5.3	Realizácia integrácie počítačového videnia do riadenia	57
6	Vizualizácia a grafické užívateľské rozhranie	62
6.1	Vizualizácia robotického ramena	62
6.2	Grafické užívateľské rozhranie	65
7	Vyhodnotenie implementovaného riešenia	69
7.1	Vyhodnotenie robotického ramena	69
7.1.1	Testovanie úchopu a presunu objektu	71
7.2	Vyhodnotenie rozpoznávania objektov	72
7.3	Vyhodnotenie skladania trojkomponentovej súčiastky	73
7.3.1	Testovanie úspešnosti skladania	74
	Zoznam príloh	81

Zoznam obrázkov

2-1	Schematická reprezentácia priamej a inverznej kinematiky	7
2-2	Priečny rez krokovým motorom s popisom jednotlivých častí	9
3-1	Senzor Microsoft Kinect v2	13
3-2	Mračno bodov transformované z hĺbkovej mapy	15
3-3	Schéma konvolúcie	21
3-4	Vstupný binárny obraz a priestor parametrov [1]	23
3-5	Transformácia hĺbkovej mapy do šedotónového obrazu	26
4-1	Hlavné okno prostredia Fusion 360	29
4-2	Porovnanie počítačného náčrtu ramena a priemyselného ramena Mitsubishi	30
4-3	Výsledný 3D návrh modelu robotického ramena	33
4-4	Schematický náčrt kinematickej štruktúry	34
4-5	Bodová vizualizácia pracovného priestoru ramena	35
4-6	Finálna fyzická podoba modelu robotického ramena	38
4-7	Robotické rameno s namontovaným úchopným mechanizmom	38
4-8	Koncepcia riadiacej skrinky	40
4-9	Schematický popis rozloženia signálov vo vodiči	41
4-10	Schéma vlastnej riadiacej dosky	42
4-11	Finálna podoba riadiacej skrinky robotického ramena	44
4-12	Schematický náčrt architektúry riadenia	45
4-13	Vývojový diagram programu riadiacej jednotky STM32	51
5-1	Stojan na senzor Kinect	53
5-2	Zložená trojkomponentová súčiastka a jej komponenty	54
5-3	Bloková schéma algoritmu rozpoznávania	55
5-4	Schéma prehľadávania hĺbkovej mapy po vrstvách	57
5-5	Hĺbková mapa (vľavo) a farebný obraz (vpravo) zo senzoru Kinect	58
5-6	Grafické znázornenie úložného priestoru	60

5-7	Okno grafického užívateľského rozhrania pre modul počítačového videnia	61
6-1	Vizualizácia ramena realizovaná knižnicou Open3D	64
6-2	Grafické užívateľské rozhranie pre robotické rameno	66
6-3	Grafické užívateľské rozhranie pripravené na použitie	68
7-1	Robotické rameno v určitej kinematickej polohe	70
7-2	Ukážka úchopu objektu mechanizmom	72
7-3	Konfúzna matica rozpoznávacieho algoritmu	73
7-4	Ukážka poskladanej trojkomponentovej súčiastky	74

Zoznam tabuliek

3-1	Technické špecifikácie senzorov Kinect	12
4-1	Vysvetlenie symbolov ku kinematickej štruktúre	34
4-2	Zoznam konfiguračných údajov pre jednotlivé aplikácie	46
5-1	Hodnoty použité pri prepočte súradníc	56
7-1	Experimentálne testovanie presnosti koncového bodu	70
7-2	Tabuľka úspešnosti úchopu objektov	71
7-3	Tabuľka úspešnosti poskladania súčiastky	75

Zoznam symbolov a skratiek

TCP	Transmission Control Protocol
SMD	Surface Mount Device
CAD	Computer-Aided Design – počítačom ošetrovaný dizajn
RGB	Red Green Blue – základné farebné zložky
ToF	Time-of-Flight
SIFT	Scale Invariant Feature Transform
OCR	Optical Character Recognition – optické rozpoznávanie znakov
ASCII	American Standard Code for Information Interchange
LoG	Laplacian of Gaussian
RANSAC	RANdom SAmples Consensus
PCB	Printed Circuit Board – doska plošných spojov
PLA	Polylactic Acid – kyselina polymliečna
NEMA	National Electrical Manufacturers Association
STL	Stereolithography – stereolitografický formát
OBJ	Wavefront 3D Object File
3MF	3D Manufacturing File
FDM	Fusion Decomposition Modelling – fúzna dekompozícia
PWM	Pulse-Width Modulation – pulzno-šírková modulácia
FTP	Foiled Twisted Pair – tienená krútená dvojlinka
I2C	Inter-Integrated Circuit – inter-integrovaný obvod, typ zbernice
UART	Universal Asynchronous Receiver-Transmitter – univerzálny asynchrónny prijímač-vysielač, typ zbernice

SPI	Serial Peripheral Interface – sériové periférne rozhranie, typ zbernice
AWG	American Wire Gauge – americká jednotka pre priemer vodiča
JSON	JavaScript Object Notation – súborový formát
WPF	Windows Presentation Foundation – grafický subsystém vyvinutý spoločnosťou Microsoft
XAML	eXtensible Application Markup Language
GUI	Graphical User Interface – grafické užívateľské rozhranie
CLI	Command Line Interface – príkazový riadok

Úvod

S narastajúcim výkonom výpočtovej techniky sa výrazne posunuli hranice možností jej uplatnenia. Postupný vývoj v oblasti mikročipov umožnil okrem iného aj vznik nových senzorov, medzi ktoré patria aj kamerové systémy. V dnešnej dobe je možné nájsť takýto systém vo viacerých oblastiach - svoje využitie nájde v autonómnych vozidlách na detekciu ostatných účastníkov cestnej premávky, či v priemysle na kontrolu kvality, triedenie produktov alebo iné priemyselné úkony. Pomocou tejto technológie je možné výrazne rozšíriť aj schopnosti robotiky, nakoľko napodobňuje optickú sústavu živočíchov a prijíma vizuálne vnemy z okolitého prostredia.

Táto diplomová práca stavia na základoch bakalárskej práce *Návrh a realizácia modelu robotického ramena so základným riadením* [2] a pokračuje tak v problematike robotických ramien a ich riadenia. Snahou tejto práce je vytvoriť robotické rameno so šiestimi stupňami voľnosti na báze krokových motorov a rozšíriť jeho riadenie o počítačové videnie. Snímané objekty budú rozpoznávané pomocou hĺbkovej mapy extrahovanej z hĺbkovej kamery, ktorú poskytuje senzor Microsoft Kinect.

Kľúčovou vlastnosťou riešenia riadenia ramena je navrhnuť ho tak, aby bolo schopné pracovať s obrazovou informáciou získavanou z kamerového systému a využiť ju na manipuláciu s objektami umiestnenými v rámci snímaného priestoru. Na identifikáciu typu objektu a zistenie jeho súradníc bude použitá hĺbková mapa poskytovaná senzorom Kinect reprezentovaná ako matica vzdialeností. Na základe týchto dát následne algoritmus vhodnými metódami počítačového videnia spracuje obraz do štvorhodnotovej informácie – typ objektu a jeho tri pozičné súradnice v priestore. Realizovaná komunikácia bude sprostredkovaná klient-server architektúrou. Klientska aplikácia počítačového videnia takto spracované údaje na vyžiadanie odošle pomocou protokolu Transmission Control Protocol (TCP) na lokálny server, kde prebehnú kinematické výpočty. Vypočítané hodnoty natočenia kĺbov sa následne odošlú do riadiacej infraštruktúry ramena, odkiaľ budú transportované cez sériovú komunikáciu do centrálnej riadiacej dosky. Rameno napokon vykoná

sériu akčných zásahov v podobe premiestňovania detegovaných objektov na vopred definovanú odkladaciu plochu, kde po splnení určitých podmienok poskladá jednoduchú trojkomponentovú súčiastku. Okrem vytvorenia fyzického modelu robotického ramena je súčasťou tejto práce aj jeho virtuálna reprezentácia v podobe vizualizácie a grafické užívateľské rozhranie s integrovanými prvkami pre ovládanie virtuálneho, ale aj fyzického modelu.

Úvod práce je venovaný analýze kinematiky robotických ramien a analýze rozpoznávania objektov pomocou hĺbkových máp. Po teoretickej stránke sú rozoberané a vysvetlené pojmy spojené s kinematikou, s metódami počítačového videnia a ich možného prepojenia do uceleného robotického systému. Ďalej je predstavená návrhová a realizačná časť robotického ramena spojená s architektúrou jeho riadenia. Kapitola návrhu a realizácie integrácie počítačového videnia do riadenia sa zameriava na stojan pre kamerový systém a architektúru algoritmu rozpoznávania. Vizualizácii a grafickému užívateľskému rozhraniu sa venuje predposledná kapitola. V závere sú výsledky práce zhrnuté vo vyhodnotení implementovaného riešenia.

1 Formulácia úlohy

Úlohou tejto diplomovej práce je navrhnuť a fyzicky zrealizovať model robotického ramena, naprogramovať jeho riadiaci softvér, vytvoriť stojan na kamerový systém Kinect a aplikovať nadobudnuté poznatky z oblasti počítačového videnia, čím sa doplní schopnosť robota reagovať na vizuálne podnety. Z uvedených dôvodov je preto potrebné vyriešiť nasledujúce úlohy:

- **Analýza kinematiky robotických ramien**

- Pri práci s robotickými ramenami je nutné poznať ich kinematiku, aby sme ich mohli využívať na vykonávanie určitých úloh. Prvý bod sa sústreďí na problematiku priamej a inverznej kinematiky potrebnej pri riadení ramien a tiež aj rozbor pohonných jednotiek kĺbov.

- **Analýza rozpoznávania objektov pomocou hĺbkových máp**

- Narozdiel od bežných farebných kamier poskytujú hĺbkové kamery o jeden priestorový rozmer navyše. V praxi to znamená viac dát, s ktorými sme schopní pracovať. Cieľom tejto analýzy je zistiť, ako hĺbkové kamery fungujú, aké metódy počítačového videnia je výhodné použiť pri spracovaní obrazu a analyzovať možnosti rozpoznávania objektov v hĺbkovej mape so zameraním sa na výpočtovo nenáročné algoritmy.

- **Návrh robotického ramena a jeho riadenia s integráciou počítačového videnia**

- Tento bod popisuje návrhovú časť šesťosého robotického ramena s implementovanými krokovými motormi s cieľom dosiahnutia vyššej presnosti a plynulejšieho chodu. Zamiera sa tiež aj na vyhotovenie návrhu konštrukcie stojanu pre Kinect. Navyše je úlohou tohto bodu aj popis návrhu hardvéru a architektúru softvéru celého systému riadenia.

- **Realizácia robotického ramena a jeho riadenia s integráciou počítačového videnia**
 - V rámci tohto bodu sú návrhové časti realizované v podobe fyzického skonštruovania robotického ramena a stojanu pre Kinect, naprogramovanie pohybov ramena, implementácia algoritmov na detekciu objektov vo vymedzenom pracovnom priestore a následná integrácia počítačového videnia do riadenia ramena. Súčasťou je aj popis vizualizácie virtuálneho modelu ramena a grafického užívateľského rozhrania.

- **Vyhodnotenie riadenia robotického ramena s integráciou počítačového videnia**
 - V záverečnej fáze bude testované samotné riadenie ramena ako aj algoritmus počítačového videnia, ktorý je zodpovedný za lokalizáciu a klasifikáciu objektov. Vyhodnocovať sa bude kinematická presnosť ramena, schopnosť uchopiť detegovaný objekt úchopným mechanizmom a presunúť ho na určené miesto a overiť presnosť rozpoznávacieho algoritmu. Posledným štádiom testovania je zostrojenie jednoduchej trojkomponentovej súčiastky.

- **Vypracovanie dokumentácie podľa pokynov vedúceho práce**
 - Posledný bod diplomovej práce sa zameriava na dokumentáciu pozostávajúcu z dvoch príručiek – používateľskej a systémovej príručky.

2 Analýza kinematiky robotických ramien

Priemyselné roboty sú v súčasnosti jedným z najprodukovanejších a najpoužívanějších typov robotov vo výrobnom priemysle, aby sa minimalizovali mzdové náklady. Mnohé odvetvia priemyslu využívajú benefity z ich používania na svoju sériovú výrobu. Priemyselné roboty boli prvýkrát nasadené približne v polovici dvadsiateho storočia. Dôvodom bolo skvalitnenie a urýchlenie výroby s cieľom značne obmedziť finančné náklady. Roboty je možné nasadzovať aj do pre človeka nepriaznivých, či dokonca nebezpečných podmienok. Robotizované manipulátory sa v praxi používajú napríklad v automobilovom priemysle na účely lakovania a oblúkového zvarovania, v prekladiskách tovaru na manipuláciu s paletami, pri výrobe plošných spojov na ukladanie Surface Mount Device (SMD) súčiastok či rôzne iné účely. Široké uplatnenie a ich užitočnosť z nich tvorí všestrannú pomoc všade tam, kde je výhodné nahradiť manuálnu ľudskú pracovnú silu. [3]

Z hľadiska riadenia je nutné spočiatku charakterizovať robota takými parametrami, aby bolo následne možné pracovať s jeho matematickým modelom. Vo všeobecnosti sa parametrický popis robota nazýva kinematická štruktúra a udáva hodnoty translačných a rotačných zložiek. Spojením týchto zložiek vzniká kinematický reťazec. Ako uvádza [4], robotov možno deliť na základe ich kinematickej štruktúry na:

- sériové roboty – otvorený kinematický reťazec (*open loop*), príkladom je robotické rameno,
- paralelné roboty – uzatvorený kinematický reťazec (*closed loop*), príkladom je Stewartova platforma,
- hybridné roboty – kombinuje obidva typy reťazcov.

Ďalším podstatným kritériom pri klasifikácii robotov je ich počet stupňov voľnosti (z angl. *Degree of Freedom*). Pojem stupeň voľnosti je definovaný ako počet parametrov potrebných na špecifikáciu konfigurácie mechanizmu, pokiaľ ide o počet

článkov, kĺbov a voľnosť pohybu v každom kĺbe [5]. V súvislosti s priemyselnými robotickými ramenami tento pojem popisuje rotačný kĺb s jednou osou otáčania. Podľa [4] v závislosti od počtu stupňov voľnosti rozdeľujeme robotov:

- univerzálny robot – so šiestimi stupňami voľnosti, jednoznačne vymedzujúci polohu a orientáciu objektu manipulácie v kartézskom súradnicovom systéme,
- redundantný robot – s viac ako šiestimi stupňami voľnosti, využívajúci väčšiu voľnosť, čím je schopný pohybovať sa v stiesnených priestoroch,
- deficitný robot – s menej ako šiestimi stupňami voľnosti, zväčša roboty typu Selective Compliance Assembly Robot Arm (SCARA).

Z uvedených pojmov je zrejmé, že priemyselné robotické rameno sa zaraďuje medzi sériové univerzálne roboty. Najviac využívaným akčným členom na pohon kĺbov je elektrický motor, pričom existuje viacero ich druhov. Podrobnejšie sa problematike kĺbovej aktuácie venuje podkapitola 2.2. Ako bolo spomenuté vyššie, priemyselné roboty používajú vo svojich kinematických reťazcoch len dva základné druhy kinematických dvojíc – rotačné (R) a posuvné (T). Podľa toho, vzhľadom ku ktorej osi súradnicového systému sa pohyb uskutočňuje sa značia rotácie ako R_x , R_y , R_z a translácie ako T_x , T_y , T_z . Na kompletne určenie polohy ľubovoľného bodu v priestore tak potrebujeme poznať 6 nezávislých súradníc.

2.1 Priama a inverzná kinematika

Dôležitou súčasťou analýzy kinematiky robotických ramien je poznať úplný kinematický model mechanického systému poskytujúci všetky podstatné veličiny ako pre dynamický model, tak aj pre potreby riadenia robotického systému. Ide najmä o priebeh polohy a orientácia koncového bodu (tiež nazývaného efektor) v čase t a tomu zodpovedajúci priebeh natočenia jednotlivých článkov celého mechanizmu. Poloha týchto článkov je všeobecne popísaná tzv. zovšeobecnenými súradnicami (v robotike sa často používa pojem *joint variables*), ktoré udávajú natočenie, či posunutie

jednotlivých pohybových osí. V rámci kinematiky sú zavedené aj dva významné pojmy – priama a inverzná kinematická úloha. Tieto pojmy boli bližšie rozoberané a popísané v bakalárskej práci [2], cieľom podkapitoly je však stručne vysvetliť základné pojmy a kľúčové znalosti na pochopenie tejto problematiky. Na obrázku 2–1



Obrázok 2–1 Schematická reprezentácia priamej a inverznej kinematiky

je symbolicky znázornená súvislosť medzi priamou a inverznou kinematikou. Uhly $\alpha, \beta, \gamma, \delta, \epsilon, \dots$ reprezentujú všetky natočenia n kĺbov kinematického reťazca a súradnice T_x, T_y, T_z a R_x, R_y, R_z predstavujú polohu koncového bodu a jeho natočenie v kartézskom súradnicovom systéme.

Priama kinematika sa zaoberá výpočtom súradníc koncového bodu kinematického zretazenia v závislosti od jeho kĺbových natočení. Príkladom môže byť robotický manipulátor, ktorý sa skladá zo sériových článkov. Tie sú k sebe pripevnené otočnými alebo prizmatickými spojmi od svojej základne až po koncový efektor. Na vyriešenie priamej kinematickej úlohy robota systematickým spôsobom je nutné použiť vhodný kinematický model [6]. Každá rotácia a každá translácia je matematicky interpretovateľná ako transformačná matica a vyjadruje jeden koordinačný rámec. Maticovým vynásobením týchto matíc sériovo za sebou dostávame homogénnu kompozitnú transformačnú maticu. Vypočítaná matica obsahuje priamu informáciu o pozícii koncového bodu kinematického reťazca, avšak na zistenie orientácie tohto bodu je vyžadovaný dodatočný prepočet [2].

Problém inverznej kinematiky je naopak podstatne zložitejší a skúma sa už mnoho desaťročí. Je nevyhnutný pri ovládaní priemyselných robotických ramien a jeho riešenie je výpočtovo náročné. Definícia inverznej kinematickej úlohy je opačná

ku priamej. Označuje spätný proces, ktorý vypočítava parametre kĺbových natočení v závislosti od špecifikovanej polohy a orientácie koncového bodu. Úlohy, ktoré má vykonávať robotický manipulátor sú definované v kartézskom priestore, zatiaľ čo aktuátory pracujú v kĺbovom priestore. Kartézsky priestor zahŕňa orientačnú maticu a polohový vektor. Kĺbový priestor je však reprezentovaný kĺbovými uhlami. Prevod polohy a orientácie koncového efektora manipulátora z karteziánskeho priestoru do kĺbového priestoru sa nazýva inverzný kinematický problém. Existujú dva prístupy riešenia, a to geometrický a algebraický, ktorý sa používa na analytické odvodenie riešenia inverznej kinematiky. [6]

Inverzné kinematické riešenia však majú určité nevýhody, napríklad riešenia nemusia existovať alebo naopak existuje viacero možných riešení, čo má za následok vyššiu výpočtovú komplexnosť napríklad keď sa zvýši počet kĺbov kinematickej štruktúry. Z tohto dôvodu sa v súčasnosti veľa výskumov pokúša nájsť novú metódu na vyriešenie problému inverznej kinematiky s menšou zložitostou výpočtov, rýchlejšiu a presnejšiu. Najpoužívanejšie metódy na riešenie inverznej kinematiky môžu byť klasifikované do troch tried – algebraická, geometrická a iteratívna metóda. Bližšie sa týmto metódam venuje [6], [7] a [2].

2.2 Pohonné jednotky kĺbov

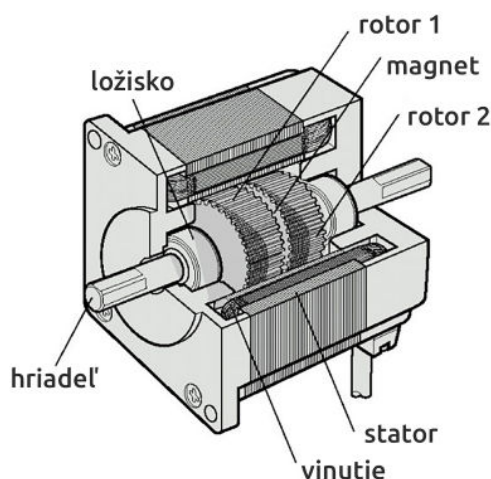
Každá riadená os robotického systému potrebuje disponovať vlastným pohonom, ktorý musí zaisťovať precízne pomalé, alebo rýchle pohyby s dostatočným zrýchlením a brzdením. Pohonná jednotka sa zvykne nazývať aktuátor a radí sa medzi akčné členy sústavy. Podľa druhu použitej energie na pohon ich delíme na:

- mechanické,
- pneumatické,
- hydraulické,
- elektrické,

- hybridné.

Spomedzi uvedených typov pohonov sa v praxi najčastejšie využívajú práve elektrické pohony. Sú najvhodnejšie pre roboty, ktoré nevyžadujú nadobúdanie vysokých rýchlostí alebo výkonu, ale vyžadujú presnosť a opakovateľnosť (z angl. *repeatability*), ako je to práve v prípade priemyselnej robotiky. Ich použitie v tomto sektore je zaujímavé najmä pre ich jednoduchú inštaláciu, ovládanie a spoľahlivosť. Elektrické pohony sú klasifikované ako motory alebo servomotory na jednosmerný prúd, motory na striedavý prúd a krokové motory.

Najpopulárnejším motorom je krokový motor (Obrázok 2–2), ktorý poskytuje vysokú presnosť a dobrú ovládateľnosť, nakoľko pracuje s individuálnymi diskretnými stavmi – krokmi. Ako uvádza Žalman [8], elektromagnetické krokové motory



Obrázok 2–2 Priečný rez krokovým motorom s popisom jednotlivých častí

sú mnohopólové a mnohofázové synchronne motory prispôsobené prevádzke v krokovom režime. Najčastejšie sú používané ako otvorené polohové číslicové servopohony, bez priameho snímania polohy rotora motora. Z konštrukčného hľadiska ich možno rozdeliť na:

- reakčné (reluktačné) motory s pasívnym rotorom – rotory majú zubové póly, medzi pólmi statora a rotora vznikajú tangenciálne elektromagnetické sily, snažiac sa pootočiť rotor tak, aby došlo k maximálnemu magnetickému toku,

- motory s permanentnými magnetmi s aktívnym rotorom – na rotore majú permanentný magnet (bezkefóvé motory) alebo budiace vinutie, na statore je n-fázové vinutie [9].

Z hľadiska vinutia cievok rozlišujeme medzi unipolárnym a bipolárnym motorom. Princípom fungovania motora je postupné prepínanie prúdových impulzov do cievok statora, čím sa vytvára točivé magnetické pole. Frekvencia spínania cievok určuje rýchlosť rotácie magnetického pola, čo sa priamoúmerne rovná otáčkam motora. V systémoch riadenia pohybu robotov sú krokové motory zvyčajne súčasťou uzatvorenej slučky, kedy dochádza ku spätnej väzbe v podobe senzorickej odpovede z enkódera umiestneného v kĺbe ramena. To má za následok vytvorenie regulačného obvodu pre daný kĺb.

Istou výzvou pri teórii riadenia robotizovaných systémov je vykonať určitý pohyb a dostať sa tak do požadovanej polohy čo možno najpresnejšie, eliminovať vibrácie a taktiež zabezpečiť rýchlostné obmedzenia v rámci vykonávaného akčného zásahu. [10]

Významnou časťou pri riešení aktuácie kĺbov v robotike sú prevody. V praxi existujú rôzne typy prevodov ako sú ozubené kolesá, ozubené hrebene, ozubené remene či reťazové prevody. Otáčky servopohonov sú redukované prostredníctvom vysoko zaťažiteľných špeciálnych prevodoviek s veľkým prevodom a pracujúcich bez vôle. Vysoké prevody umožňujú získať veľké krútiace momenty potrebné pre pohyby ramien robotov. [9]

3 Analýza rozpoznávania objektov pomocou hĺbkových máp

Okrem konvenčnej farebnej kamery (podobné ľudskému videniu) existuje aj špeciálny druh, ktorý je schopný pracovať aj s informáciou o vzdialenosti medzi kamerou a snímaním prostredím. Takýmto prostriedkom je hĺbková kamera. Umožňuje získať dáta z rozmeru navyše, čo jej dáva výhodu najmä pri komplexnejších úlohách, kde už nepostačuje bežný kamerový systém. V mobilnej robotike sa používa napríklad na zisťovanie vzdialeností okolitých objektov od robota za účelom vyhýbania sa prekážkam či na mapovanie prostredia. Uplatňuje sa však aj pri detekcii a rozpoznávaní objektov za účelom ich manipulácie, na čo sa zameriava aj táto diplomová práca.

Táto kapitola popisuje niektoré z metód počítačového videnia, na základe ktorých je možné problematiku rozpoznávania objektov umiestnených v rámci určitého vymedzeného pracovného priestoru riešiť. Vzhľadom k tomu, že diplomová práca je postavená hĺbkovej kamere senzoru Kinect bude súčasťou analýzy rozpoznávania aj predstavenie tohto zariadenia.

3.1 Microsoft Kinect V2

V roku 2010 spoločnosť Microsoft poprvýkrát ukázala svetu nový druh senzoru. Kinect bol navrhnutý tak, aby slúžil ako interaktívny ovládač pre konzoly Xbox. Skladá sa z kamerového systému pozostávajúceho z dvoch častí – farebnej Red-Green-Blue (RGB) kamery a laserového projektora operujúceho v infračervenom spektre. Prvá generácia senzoru Kinect využívala na snímanie hĺbky metódu tzv. *štruktúrovaného svetla*. Spoliehala sa teda na výpočet hĺbky na základe projekcie istého infračerveného vzoru. Naproti tomu, novšia druhá verzia Kinect-u vydaná v roku 2012 používa na zachytenie informácie o hĺbke iný prístup – technológiu *Time-of-Flight* (ToF). Princíp fungovania je založený na vysielaní infračervených lúčov do prostredia a me-

ria sa čas, za ktorý sa po odraze od povrchov tieto lúče spätne zachytia. To rapídne zvyšuje presnosť tohto senzoru oproti jeho predchodcovi a napomáha tak širokému využitiu na praktické účely nielen v hernom priemysle. Okrem zlepšenia hĺbkovej kamery došlo k zvýšeniu rozlíšenia RGB kamery, ktorá už disponuje Full HD kvalitou obrazu. [11] V uvedenej tabuľke 3–1 sú pre porovnanie zobrazené technické para-

Parameter	Kinect V1	Kinect V2
RGB kamera	640 × 480 px	1920 × 1080 px
Frekvencia snímania	30 fps	30 fps
Hĺbková/IR kamera	320 × 240 px	512 × 424 px
Max. vzdialenosť	~ 4500 mm	~ 4500 mm
Min. vzdialenosť	400 mm	500 mm
Technológia	<i>Structured Light</i>	<i>Time-of-flight</i>
Motorizovaný podstavec	áno	nie

Tabuľka 3–1 Technické špecifikácie senzorov Kinect

metre oboch verzií senzoru Kinect. Štúdií popisujúcich rozdiely a porovnania týchto dvoch verzií je mnoho. Napríklad Wasenmüller [12] uvádza vo svojej knihe využitie tejto technológie v oblasti robotiky a počítačového videnia. Systematicky porovnáva obe verzie z hľadiska presnosti a precíznosti. Zameriava sa aj na vznik možných snímacích chýb a odchýliek, ako sú *lietajúce pixely* (z angl. *flying pixels*) či *multitrasová interferencia* (z angl. *multipath interference*). Vzhľadom k tomu, že táto práca využíva na získavanie hĺbkovej mapy práve druhú verziu Kinect-u, bližšie sú analyzované jej technické detaily.

Ako už bolo spomenuté v úvode, Kinect V2 (Obrázok 3–1) je založený na ToF princípe a ponúka vyššie rozlíšenie hĺbkovej mapy a širšie zorné pole ako jeho predchodca. Keďže z fyzikálneho hľadiska sa jedná o meranie času medzi vyslaním a prijatím elektromagnetického signálu, je možné tento senzor používať v interiéri, ale aj v exteriéri. Avšak takýto princíp získavania hĺbkovej mapy je náchylný na vznik odchýlok kvôli určitým faktorom prostredia (odrazy, nerovnosť povrchov, typ po-

vrchu a podobne). Pri snímaní môže dôjsť ku nesystematickým chybám, pričom najčastejšou z nich je šum. Podľa [13] existujú pri tomto senzore dva druhy šumu – axiálny a laterálny šum. Axiálny šum zapríčiňujú rozdiely v meraní času pozdĺž osi z . Pri tomto type šumu teda dochádza ku nepresným meraniam tretej rozmerovej súradnice. Laterálny šum zase spôsobujú rozdiely v pixeloch pozdĺž hrán v rámci roviny hĺbkového obrazu. V praxi to znamená nepresnosť hrán snímaných objektov vo výslednom hĺbkovom obraze.



Obrázok 3 – 1 Senzor Microsoft Kinect v2

Na to, aby bolo možné nasadiť senzor do aplikácie počítačového videnia je nutné spracovať obrazové informácie v počítači. Nakoľko Kinect je primárne perifériou pre konzoly Xbox, spoločnosť Microsoft prišla na trh so špeciálnym adaptérom umožňujúcim pripojiť Kinect do počítača s operačným systémom Windows cez univerzálnu sériovú zbernicu (USB). Podrobnosti pripojenia a konfigurácie sú uvedené v dokumentácii tejto práce. Pracovať s obrazovou informáciou je napokon realizovateľné vo viacerých programovacích jazykoch. Knížnice na manipuláciu s dátami produkovanými Kinect-om poskytuje C#, Python alebo Java.

3.2 Hĺbková mapa

Pred zavedením pojmu hĺbkovej mapy je potrebné najprv spomenúť, ako je interpretovaný farebný obraz z bežnej kamery (napr. fotoaparát v smartfóne). Každý farebný obrázok je v počítači digitálne reprezentovaný ako dvojrozmerná matica pi-

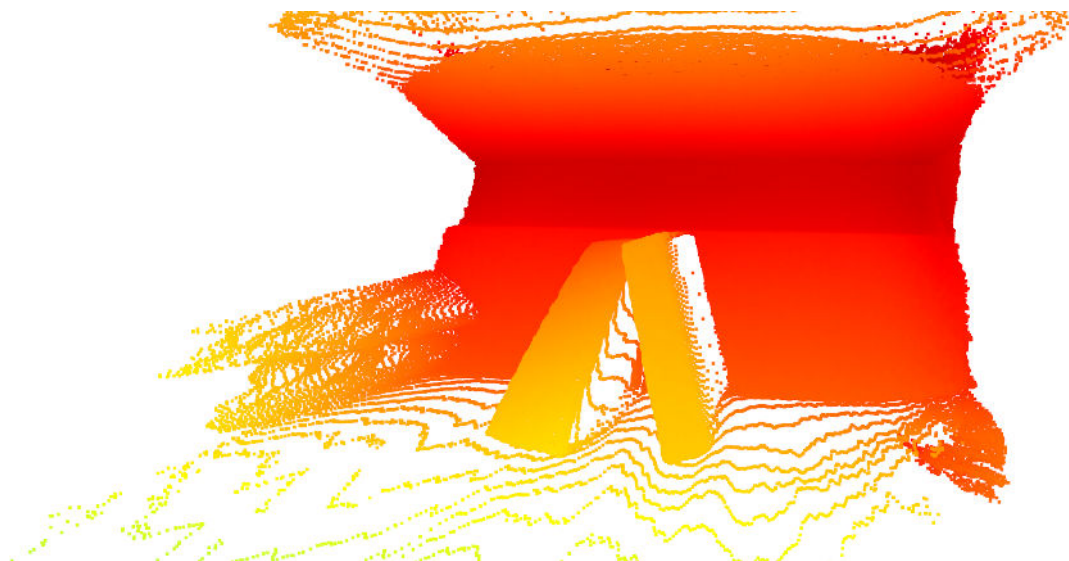
xelov. Pixel je najmenší zobrazovací bod v počítačovej rastrovej grafike. Pozostáva z trojice hodnôt, pričom každá hodnota prislúcha jednej základnej farebnej zložke, ktorú nazývame farebný kanál. Spojením kanálov pre červenú, zelenú a modrú farbu potom vzniká celková farba pixelu. Matica \mathbf{I} (3.1) zjednodušene reprezentuje obrázok ako mriežku pixelov s rozmermi $m \times n$.

$$\mathbf{I} = \begin{bmatrix} [r, g, b]_{11} & [r, g, b]_{12} & \cdots \\ [r, g, b]_{21} & \ddots & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix} \quad (3.1)$$

Naproti tomu hĺbková mapa predstavuje dvojrozmernú maticu hodnôt vzdialeností (3.2) medzi snímacou plochou hĺbkovej kamery a snímaným povrchom. Senzor Kinect meria túto vzdialenosť v milimetroch pričom nadobúda hodnoty z diskrétneho rozsahu $400mm \leq d \leq 4500mm$. Je však nutné podotknúť, že rozmernosť takto získanej hĺbkovej mapy je len tzv. 2.5D, vzhľadom k tomu, že sa jedná len o projekciu povrchových bodov a nezískavajú sa tak plnohodnotné volumetrické dáta.

$$\mathbf{D} = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ d_{m1} & \cdots & \cdots & d_{mn} \end{bmatrix} \quad (3.2)$$

Ako uvádza Haladová a kol. [14], k informáciám z hĺbkovej mapy môžeme pristupovať aj inak, a to prostredníctvom projekcie do tzv. mračna bodov (z angl. *pointcloud*) (Obrázok 3–2). Takýto prístup zaručuje prípadnú vizualizáciu dát v podobe izolovaných bodov v 3D priestore. Keďže Kinect sníma RGBD dáta (3 farebné kanály červenú, zelenú, modrú a hĺbku), umožňuje to každému bodu navyše priradiť aj farebnú hodnotu. Výhodou transformácie hĺbkovej mapy do mračna bodov je ich použiteľnosť v algoritmoch ako je napr. 3D Scale Invariant Feature Transform (3D SIFT) na detekciu kľúčových bodov a extrakciu deskriptorov, čo sa hodí najmä pri pokročilejších úlohách počítačového videnia.



Obrázok 3 – 2 Mračno bodov transformované z hĺbkovej mapy

3.3 Metódy spracovania obrazu

Vo všeobecnosti každú obrazovú informáciu získavanú prostredníctvom snímacích zariadení je potrebné predspracovať. Z hľadiska teórie informácie v priebehu predspracovania nedochádza ku strate informácií, tie sa môžu len zvýrazniť alebo potlačiť. Je to spôsobené najmä tým, že zosnímaný obraz obsahuje nadbytočné údaje a je ich možné pomocou určitých metód zredukovať. Medzi najčastejšie odstraňované poruchy obrazu patrí šum (biely, Gaussov, aditívny, ...), skreslenie dané vlastnosťami snímacieho zariadenia (korekcia pomocou afinných transformácií) [15], korekcia jasů a kontrastu. Po predspracovaní je následne možné aplikovať rôzne druhy metód resp. algoritmov, ktoré umožňujú obraz ďalej spracovať. Zaraďujeme k nim prahovanie, morfológické operácie (dilatácia, erózia), detekciu hrán (Cannyho detektor hrán), konvolúciu, interpoláciu ale aj mnohé ďalšie. Táto podkapitola je venovaná špecificky metódam, ktoré napomáhajú spracovať hĺbkovú mapu získanú zo senzoru Kinect za účelom získania kľúčových informácií. Za pomoci uvedených metód sme tak schopní zistiť polohu objektov, ich základné črty a hrany, ale tiež aj rovnicu popisujúcu rovinu pod objektmi. Týmto spôsobom sme vieme navyše zistiť výšku, v akej sa relatívne voči tejto rovine najvyšší objekt nachádza.

3.3.1 Prahovanie a morfológické operácie

Podľa [16] sú segmentačné algoritmy založené jedným z dvoch základných vlastností diskontinuity hodnôt intenzity a podobnosti. Prvou kategóriou je rozdelenie obrazu na základe náhlych zmien intenzity, akými sú napríklad hrany. Druhá kategória je založená na podelení obrazu do suboblastí, ktoré sú si podobné podľa vopred definovaných kritérií. Nech $f(i, j)$ je vstupným obrazom, potom prahovaním prahom \mathbf{T} dostaneme výstupný obraz $g(i, j)$ (3.3) nasledovne:

$$g(i, j) = \begin{cases} 1 & \text{pre } f(i, j) > T \\ 0 & \text{pre } f(i, j) \leq T \end{cases} \quad (3.3)$$

Prahovacie techniky segmentácie môžu byť vo všeobecnosti rozdelené do troch tried:

- lokálne – založené na vlastnostiach pixelov a ich lokálnych susedstiev, hodnota prahu \mathbf{T} sa mení v závislosti od lokálnych podmienok,
- globálne – založené na vlastnostiach obrazu ako celku (napr. histogram obrazu), hodnota prahu \mathbf{T} je konštantná,
- automatické – využívajúce aj lokálne aj globálne vlastnosti obrazu.

Prahovanie je najjednoduchšou formou segmentácie obrazu vzhľadom k výpočtovej náročnosti tejto operácie. Využíva sa najmä pri šedotónovom obraze, ale je možné ho aplikovať aj na jednotlivé farebné kanály RGB obrazu. Výstupom prahovania je binárny obraz – matica obsahujúca hodnoty 0 alebo 1. Táto metóda sa využíva napríklad pri Optical Character Recognition (OCR) technike na prevod tlačených či písaných znakov do počítačom interpretovateľnej podoby (napr. ASCII znaky).

S prahovaním súvisia aj ďalšie metódy spracovania obrazu. Patria sem morfológické operácie. Spôsobujú, že po ich aplikovaní na obraz zvýrazňujú, resp. potláčajú určité črty. Názov vznikol z inšpirácie biologickej morfológie a vyjadruje vizuálnu, či inú zmenu z jedného stavu do druhého. Matematická morfológia sa vo všeobecnosti

zaoberá štúdiom objektov z hľadiska ich tvaru a geometrie. Využíva pri tom kombinácie metód z teórie množín, topológie a diskretnej matematiky. Preto je možné ju použiť aj v spracovaní obrazu či ďalších diskretných dát. Podľa knihy [14] morfológické spracovanie obrazu využíva informáciu o susedných pixeloch v topologickom okolí spracovávaného pixelu. Primárne boli definované ako operácie pre binárne obrazy. Rozšírením sú morfológické operácie pre šedotónové či dokonca viacnásobné (farebné) obrazy. Základným pojmom pri morfológii je štruktúrálny element, čo je matica s nepárnou rozmernosťou, aby sa zabezpečilo, že práve jeden pixel je stredom štruktúrného elementu. Tento bod je nazývaný kotevným bodom (z angl. *anchor point*). Typicky je to matica pozostávajúca iba z jednotiek a núl, príklad štvorcového elementu je matematicky zapísateľná ako (3.4).

$$\mathbf{S} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (3.4)$$

Okrem toho však môžu existovať aj iné tvary, ako je napríklad elipsa, krížik alebo iný ľubovoľný tvar (napr. v tvare písmena L a podobne). Rozmery štruktúrného elementu sú spravidla podstatne menšie než vstupný obraz. Morfológickú transformáciu môžeme definovať ako matematickú reláciu medzi dvomi bodovými množinami – množinou obrazu a množinou štruktúrného elementu. Medzi základné transformácie patria:

- dilatácia,
- erózia,
- morfológické otvorenie (kombinácia erózie a následnej dilatácie s použitím rovnakého štruktúrného elementu),
- morfológické uzavretie (kombinácia dilatácie a následnej erózie s použitím rovnakého štruktúrného elementu).

Binárna dilatácia zväčšuje plochu bieleho objektu na čiernom pozadí, binárna erózia vykonáva presný opak – znižuje plochu bieleho objektu na čiernom pozadí. Pri invertovaných farbách dosiahneme samozrejme analogicky opačný efekt. Matematicky by sme vedeli tieto operácie zapísať ako (3.5) a (3.6):

$$F \oplus S \quad (\text{dilatácia}) \qquad F \ominus S \quad (\text{erózia}) \qquad (3.5)$$

$$(F \ominus S) \oplus S \quad (\text{otvorenie}) \qquad (F \oplus S) \ominus S \quad (\text{uzavretie}) \qquad (3.6)$$

kde F je vstupný obraz a S je použitý štruktúrny element. Uvedené morfológické operácie je výhodné použiť napríklad za predpokladu, že vstupný binárny obraz podlieha zrnitosti. Aplikovaním dôjde k odstráneniu zrn bez výraznejšieho narušenia hlavných oblastí. Nakoľko je z hĺbkovej mapy možné extrahovať binárny obraz pomocou prahovania výškou, môžu sa morfológické operácie použiť na odstránenie falošných fragmentov spôsobených šumom senzoru.

3.3.2 Detekcia hrán

Veľmi dôležitou a široko zaužívanou súčasťou spracovania obrazu je detekcia hrán. Zaraduje sa medzi netriviálne úlohy a vo svojej podstate ide o nájdenie takých pixelov na obraze, ktoré opisujú náhle zmeny jasovej zložky. Jedná sa o diskontinuitu medzi dvoma odlišnými regiónmi. Matematicky sú tieto diskontinuity opísateľné ako gradienty – zmeny hodnôt jasu v rôznych smeroch v závislosti od počtu premenných. Rozlišujeme dva základné metódy využitia derivácií za účelom nájdenia hrán. Metóda, pri ktorej sa používa prvá derivácia jasovej funkcie sa spolieha na nájdenie extrémov, konkrétne maxím. Existuje skupina operátorov pričom každý operátor má svoje vlastné konvolučné jadro. Patrí sem Sobelov, Prewittov, Kirschov, Robinsonov a Robertsonov operátor. Z hľadiska rozmernosti je obrazová funkcia funkciou dvoch premenných $f(x, y)$, preto je gradient pre dvojrozmerný signál definovaný ako vektor prvých parciálnych derivácií (3.7).

$$\nabla f(x, y) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right) \qquad (3.7)$$

Keďže gradient je vyjadrený vektorovo, má okrem veľkosti aj svoj smer. Je orientovaný v smere najvyššieho nárastu intenzity a jeho veľkosť zodpovedá rýchlosti zmeny jas. Takto je možné určiť, ktorým smerom sa intenzita jas mení. Hrany rozdelujeme podľa typu na:

- skokovú hranu,
- nábežnú hranu,
- impulznú hranu.

Pri spojitaj obrazovej funkcii sú zložky gradientu definované deriváciami v horizontálnom a vertikálnom smere. V prípade, že zmena v jednom z týchto dvoch smerov je nulová, gradient sa prejavuje len v nenulovom smere. Druhou metódou je založiť detekciu hrán na druhej derivácii. Tá využíva na hranovú detekciu prechod druhej derivácie cez nulu (z angl. *zero-crossings*). Matematicky je takýto prístup vyjadriteľný ako (3.8):

$$\nabla^2 f = \left(\frac{\partial^2 f}{\partial x^2}, \frac{\partial^2 f}{\partial y^2} \right) \quad (3.8)$$

čo je Laplaceov operátor. Druhá derivácia je však omnoho citlivejšia na šum a preto sa zväčša používa v kombinácii s dodatočným vyhladzovaním. Jedným z možných riešení je skombinovať Laplacián s Gaussovským rozostrením, čím vznikne takzvaný Laplacian-of-Gaussian (skrátene LoG) (3.9). Rozostrenie obrazu spôsobí redukciu šumu, čím vytvorí robustnejšie riešenie hranovej detekcie.

$$\nabla^2 [G(x, y, \sigma) * f(x, y)] \quad (3.9)$$

V digitálnom obraze je intenzita známa len v diskretných bodoch. Derivácie nie je teda možné definovať bez predpokladu existencie spojitaj obrazovej funkcie. Aproximácie derivácií obrazovej funkcie sa najčastejšie vykonávajú pomocou konvolúcie obrazu napr. so Sobelovým alebo Prewittovým operátorom.

Hranový detektor je algoritmus, ktorý vyhodnocuje jasové zmeny v obraze a snaží sa zvýrazniť hrany – pixely na rozhraní diskontinuit. Detektor hrán prijíma diskrétnu, digitalizovanú obrázku ako vstup a vytvára mapu hrán ako výstup. Hranová mapa niektorých detektorov obsahuje explicitné informácie o polohe a sile hrán, ich orientácii a mierke. Počas historického vývoja tohto druhu spracovania obrazu boli vyvinuté rôzne detektory, ktoré sa líšia svojim účelom (fotometrické a geometrické vlastnosti hrany) a ich matematické a algoritmické vlastnosti. Najznámejším a veľmi populárnym detektorom je v súčasnosti Cannyho hranový detektor a má široké praktické využitie v rôznych oblastiach. [17]

3.3.3 Konvolúcia

Ďalšou podstatnou súčasťou spracovania obrazu je konvolúcia. Zaraďuje sa medzi lineárne filtrácie v priestorovej oblasti, nakoľko sa nezameriava na manipuláciu hodnôt vo frekvenčnej, ale v priestorovej doméne. V praxi sa využíva v mnohých oblastiach od konvolučných neurónových sietí na rozpoznávanie objektov na fotografii až po filtre používané sociálnymi sieťami. Jedná sa o matematický operátor (značí sa $*$) medzi dvomi funkciami $f(x)$ a $g(x)$ (3.10). Konvolúcia je definovaná ako integrál vyjadrujúci veľkosť prekrytia funkcie $f(x)$ pri jej posunutí cez inú funkciu $g(x)$, čím plynule takpovediac “zmieša“ jednu funkciu s druhou [18].

$$(f * g)(x) = \int_{-\infty}^{\infty} f(\tau)g(x - \tau) d\tau \quad (3.10)$$

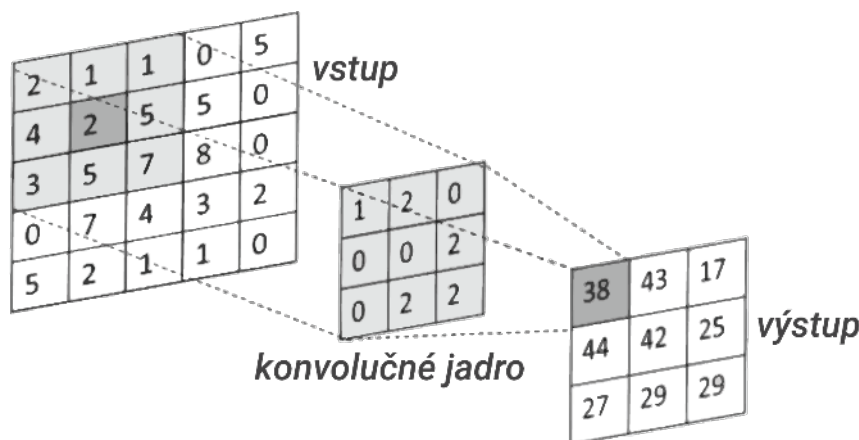
Jednou z vlastností operácie konvolúcie je komutatívnosť, čo znamená, že nezáleží na poradí funkcií. V tejto časti podkapitoly sa nebudeme zaoberať matematickými podrobnosťami, skôr sa zameriame na jej aplikácie v rámci práce s dvojrozmernou obrazovou informáciou. Výpočet konvolúcie prebieha ako postupnosť jasne definovaných základných matematických operácií. Počítačová implementácia výpočtu bola v priebehu vývoja výpočtovej techniky optimalizovaná tak, aby sa skrátil čas potrebný na výpočet tejto operácie. Základom pre prácu s konvolúciou je definovanie filtračného resp. konvolučného jadra, čo je matica $m \times n$ spravidla štvorcová s nepár-

nou rozmernosťou strany (napríklad 3×3 , 11×11 , ...). To zaručuje, že je jednoznačne definovaný stred tejto matice práve jedným prvkom. Tento prvok je dôležitý z hľadiska priebehu výpočtu. Pre každú posunutú pozíciu konvolučného jadra počítame sumu súčinov hodnoty konvolučného jadra a príslušného pixelu obrazu. [14]

Konvolúcia je prítomná najmä pri algoritmoch spracovania dvojrozmerných diskretných obrazových funkcií v počítačovej grafike. Jej diskretná forma preto môže byť zapísaná ako (3.11):

$$(f * h)(x, y) = \sum_{i=-k}^k \sum_{j=-k}^k f(x - i, y - j) \cdot h(i, j) \quad (3.11)$$

Má široké aplikačné využitie pri detekcii hrán, vyhladzovaní či rozmazaní obrazu, odstraňovaní šumu a podobne. Na obrázku 3–3 je znázornená zjednodušená schéma priebehu výpočtu konvolúcie s použitím konvolučného filtra o rozmeroch 3×3 . Z ob-



Obrázok 3–3 Schéma konvolúcie

rázku je zrejmé, že výpočet okrajových hodnôt je problematický, pretože ak priložíme konvolučné jadro k okrajovému pixelu vstupného obrazu, výpočet konvolúcie je neúplný v dôsledku chýbajúcich hodnôt za okrajmi. Z toho dôvodu sa v praxi používa niekoľko stratégií na vyriešenie tohto problému:

- doplnenie chýbajúcich hodnôt pomocou nulových hodnôt (tzv. ponorenie celého obrazu do “mora núl“),

- doplnenie chýbajúcich hodnôt zrkadlením okrajových pixelov obrazu,
- zmenšenie výstupného obrazu, najmenej vhodný spôsob, pretože sa mení veľkosť výstupného obrazu.

Najčastejšie sa stretujeme s aplikovaním prvého alebo druhého spôsobu, ale tieto prístupy musia zohľadniť nerelevantnosť okrajových pixelov po konvolúcii. To ale v praxi nespôsobuje problémy, nakoľko rozmery jadra sa vyberajú pokiaľ možno malé.

3.3.4 Houghova transformácia

Jedným z prístupov, ako z obrazu extrahovať parametricky opísateľné tvary je Houghova transformácia. Pomenovaná je po Paulovi Houghovi, ktorý si túto metódu patentoval v roku 1962 a svoje využitie našla v oblasti obrazovej analýzy, počítačom videní a digitálnom spracovaní obrazu. Neskôr sa táto metóda upravovala a do stavu, v akom ju poznáme dnes vďačíme dvojici Richard Duda a Peter Hart [19]. Tí ju rozšírili na všeobecnú Houghovu transformáciu schopnú identifikácie ľubovoľných tvarov a kriviek. Najčastejšie sa však používa na detekciu čiar, kruhov, poprípade elíps. Vstupom do metódy je binárny obraz získaný z hranového detektora. Vzhľadom k tomu, že vstup je diskretný, budú existovať isté tvarové odchýlky v dôsledku šumu, nedokonalosti detegovaných hrán a podobne. Houghovu transformáciu si ilustrujeme na príklade hľadania priamok, čo je najjednoduchšou formou detekcie. Na jednoznačné určenie priamky nestačí poznať súradnice jedného bodu, ale potrebujeme poznať aspoň dva. Vychádzame zo smernicovej rovnice priamky $y = ax + b$ vyjadrenej v polárnych súradniciach (3.12):

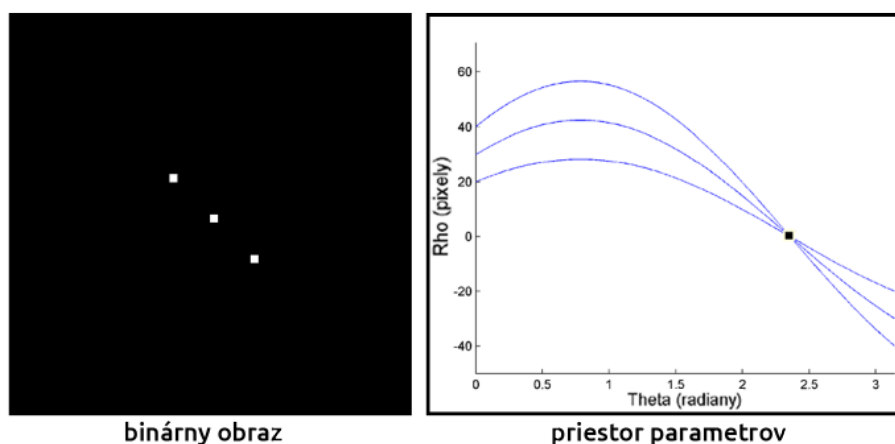
$$y = -\frac{\cos \theta}{\sin \theta}x + \frac{r}{\sin \theta} \quad (3.12)$$

Po úprave na vhodnejší tvar dostaneme (3.13):

$$r = x \cos \theta + y \sin \theta \quad (3.13)$$

kde r je vzdialenosť normály priamky od počiatku súradnicového systému a θ je uhol medzi kladnou časťou osi x a normály [20]. Takto definovaná priamka je opísateľná

dvoma parametrami r a θ , čo je reprezentované bodom v priestore parametrov (niekedy označovaného ako Houghov priestor). Výhodou prístupu je lineárny a konečný rozsah týchto dvoch parametrov, nevznikajú problémy s divergenciou. Houghova transformácia vezme každý biely pixel binárneho obrazu a reprezentuje ho ako sínusoidu v priestore parametrov. Tým sa získa množina kriviek v Houghovom priestore, ktoré sa za predpokladu, že sa priamka v obraze nachádza, pretínajú. Grafická reprezentácia transformácie binárneho obrazu do priestoru parametrov je zobrazená na obrázku 3–4. Je však nutné spojité krivky previesť do ich diskretného tvaru,



Obrázok 3–4 Vstupný binárny obraz a priestor parametrov [1]

keďže pracujeme s diskretným vstupným obrazom. Diskretizovaný tvar je reprezentovaný akumulátorom (maticou), kde prebieha hlasovanie o najpravdepodobnejšie parametre priamky. Úloha hľadania priesečníka spojitých kriviek je tak redukovaná na úlohu hľadania maxima matice akumulátora. Priesečníkom je bod s najvyšším dosiahnutým počtom hlasov a extrakcia parametrov sa stáva veľmi jednoduchou.

Podobným spôsobom vieme detegovať aj kružnice s rozdielom odlišného počtu parametrov definujúcich geometrický tvar, z čoho vyplýva aj odlišná rovnica (3.14):

$$r^2 = (x - c_x)^2 + (y - c_y)^2 \quad (3.14)$$

Kružnica je popísaná polomerom r , a súradnicami jej stredu c_x , c_y . Je nutné poznamenať, že pri výpočte uvažujeme nad fixným polomerom, nakoľko každý polomer má

vytvorený svoj vlastný akumulátor. Algoritmus prechádza všetkými bielymi pixelmi binárneho obrazu a vytvára v Houghovom priestore diskretnú kružnicu s vopred daným polomerom r okolo každého z nich. Na záver sa rovnako ako pri detekcii priamky nájde maximum akumuláčnej matice, čím sa získa pozícia stredu detegovanej kružnice s určeným polomerom r .

V súčasnosti sú spomenuté typy detekcií implementované v knižniciach pre rôzne programovacie jazyky optimalizované z výpočtového hľadiska, napríklad v knižnici OpenCV pre jazyky Python a C++. Táto knižnica poskytuje kompletnú parametrizáciu postupu detekcie. Vstupom do funkcie je šedotónový obraz spoločne s parametrami, ako sú napríklad parametre interného Cannyho detektora hrán, minimálny alebo maximálny polomer kružnice, respektívne dĺžky úsečky, alebo minimálna vzdialenosť medzi nájdenými útvarmi.

3.3.5 RANSAC

Random Sample Consensus (skrátene RANSAC) je typ algoritmu lineárnej regresie. Bol predstavený v roku 1981 [21] a poskytuje pomerne spoľahlivé určenie lineárnej funkcie na základe vstupnej vzorky dát. Tento principiálne jednoduchý algoritmus je robustný a odolný voči tzv. odľahlým hodnotám (z angl. *outliers*). V praxi to znamená zistenie takej lineárnej funkcie, ktorá optimálne popisuje vstupné dáta. V geometrickom kontexte sa využíva v 2D rovine na zistenie rovnice priamky alebo 3D priestore na zistenie rovnice roviny. Stručný popis algoritmu by bolo možné spísať do štyroch základných bodov:

1. Zo vstupnej dátovej vzorky sa náhodne vyberie n bodov (na jednoznačné určenie priamky sú potrebné dva body, na jednoznačné určenie roviny tri body).
2. Vypočítajú sa parametre lineárnej funkcie prechádzajúcej týmito bodmi.
3. Vyberie sa parameter δ vyjadrujúci prah vzdialenosti od lineárnej funkcie nájdenej v predošlom bode. Algoritmus spočíta tie dátové body, ktoré spĺňajú prahovú podmienku. Počet bodov považujeme za skóre tejto funkcie.

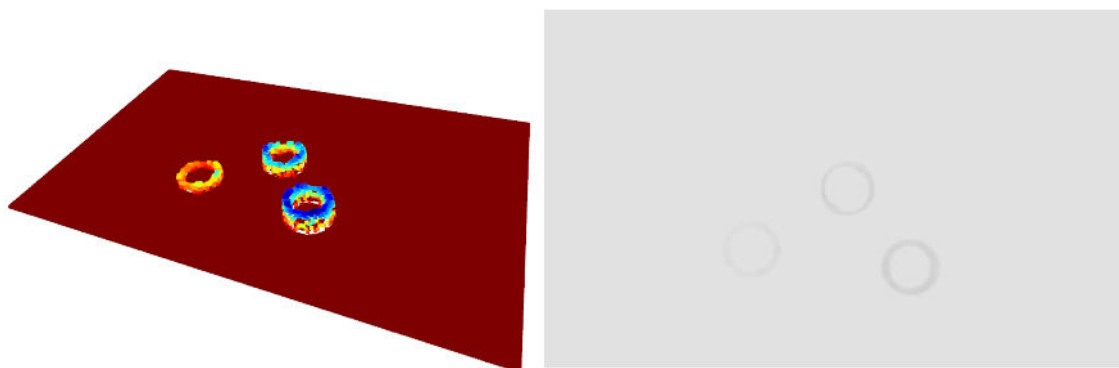
4. Postup sa cyklicky opakuje určitý fixný vopred zadaný počet iterácií.

Po ukončení cyklu algoritmus vracia lineárnu funkciu s najvyšším počtom bodov ležiacich v okolí funkcie (z angl. *inliers*). RANSAC sa používa v počítačovom videní na spájanie obrázkov na základe kľúčových bodov alebo na redukciu nesprávnych asociačných kľúčových bodov využívaných v metódach ako je SIFT. Svoje využitie ale nájde aj v súvislosti s hĺbkovými kamerami na určenie rovnej plochy, čo je výhodou najmä pri práci so segmentáciou popredia a pozadia na základe hĺbky.

3.4 Rozpoznávanie objektov

Spojením viacerých metód spracovania obrazu vieme dostať algoritmus umožňujúci riešiť komplexnejšie problémy. Jedným z nich je aj rozpoznávanie objektov. Cieľom rozpoznávania je detegovať a jednoznačne identifikovať objekt z vopred definovaného etalónu. Pre pochopenie princípu si uvedieme jednoduchý príklad – majme dvojicu dvojrozmerných geometrických útvarov (trojuholník a štvorec) položených na ploche, ktorú sníma kamera. Distorziu spôsobenú optickou sústavou kamery v tomto prípade zanedbávame. Pomocou základných metód spracovania obrazu si prevedieme vstupný RGB obraz do šedotónového spektra, odkiaľ vieme s obrazovou informáciou výpočtovo výhodnejšie pracovať, nakoľko sa jedná už len o dvojrozmernú maticu hodnôt jasov. Následne predspracujeme tento obraz hranovým detektorom, čím ho prevedieme na binárny obraz zvýrazňujúci hrany útvarov. Poslednou časťou rozpoznávania je aplikovať lineárnu Houghovu transformáciu, ktorého výstupom sú parametre nájdených priamok. Ak algoritmus nájde práve tri priamky, vieme s určitosťou povedať, že na obraze sa nachádza trojuholník, v opačnom prípade štvorec.

Podobným prístupom sme schopní vyhodnocovať aj hĺbkové mapy. Jediným rozdielom je využívanie nielen hĺbkovej mapy, ale aj jej šedotónovej podoby prepočítaním jednotlivých hodnôt vzdialeností na hodnoty jasov pixelov. Pre lepšiu predstavu je na obrázku 3–5 ilustrovaná paralela medzi hĺbkovou mapou reprezentovanou ako mračno bodov a šedotónovým obrázkom. Objekty vystupujúce z roviny



Obrázok 3–5 Transformácia hĺbkovej mapy do šedotónového obrazu

homogénneho povrchu (znázornené tmavočervenou farbou) nadobúdajú nižšie jasové hodnoty pixelov ako ich okolie. Týmto spôsobom je možné dostať šedotónový obraz vhodný na predspracovanie konvenčnými metódami popísanými v predošlej podkapitole. Výhodou riešenia je získanie vyššieho množstva informácií na podporu rozpoznávania. S takto spracovaného obrazu napokon extrahujeme informácie v podobe počtu nájdených priamok, kružníc s určitým polomerom, počtu blobov alebo rôznych iných čít. Ak vezmeme do úvahy princíp segmentácie obrazu a aplikujeme ho na hĺbkovú mapu, môžeme prahovanie realizovať nie na jasovej zložke, ale na základe hodnôt hĺbky.

4 Návrh a realizácia robotického ramena a jeho riadenia

Kapitola opisuje proces tvorby návrhu modelu robotického ramena so šiestimi stupňami voľnosti. Zameriava sa na konštrukčné aspekty, pohonný mechanizmus, riadiace elektronické komponenty a tiež aj schematické zapojenie obvodov riadenia spolu s realizáciou uvedených častí.

4.1 Návrh a realizácia konštrukčnej časti robotického ramena

Aby sme boli schopní úspešne navrhnuť a následne aj zrealizovať navrhované riešenie, je potrebné vybrať správne nástroje a ovládať ich do takej miery, aby sa zabezpečil očakávaný výsledok. V prípade 3D návrhu všetkých dielov pre robotické rameno je najlogickejšou voľbou taký typ softvéru, ktorý je primárne určený na tvorbu 3D modelov. Naopak predpríprava, ako aj samotná realizácia si vyžaduje iný, špecializovaný softvér na prípravu navrhnutých súčiastok pre 3D tlač spoločne s fyzickým zariadením – 3D tlačiarňou. Využitím týchto prostriedkov sa realizujú vopred pripravené komponenty do fyzickej podoby.

4.1.1 Nástroj Autodesk Fusion 360

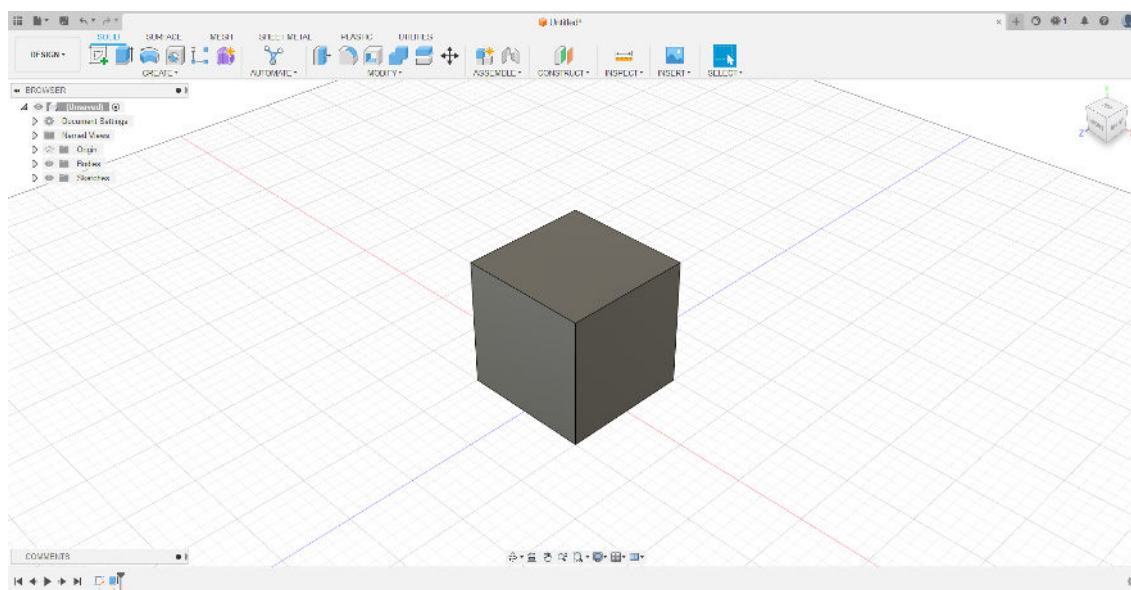
Na to, aby sme v počítači boli schopní vytvárať a vo všeobecnosti manipulovať s 3D objektmi, je potrebné naučiť sa pracovať so softvérovými nástrojmi na to určenými. Najrozšírenejšou technológiou je počítačom podporovaný dizajn (z angl. *Computer-Aided Design* – CAD), pričom na trhu jestvuje množstvo programov poskytujúcich užívateľovi prácu s 3D počítačovou grafikou. Patrí k nim aj jednoduchý nástroj Tinkercad. Bol použitý na vymodelovanie 3D komponentov robotického ramena v bakalárskej práci [2]. Medzi profesionálne programy sa radí softvér od spoločnosti Autodesk s názvom Autodesk Fusion 360. Jedná sa o polyfunkčný nástroj na tvorbu

dizajnov a ich renderov, prípravu výrobných procedúr, vytváranie elektronických schém a plošných spojov (z angl. *Printed Circuit Boards* – PCB), tvorbu simulácií a v neposlednom rade aj vytváranie animácií navrhnutého dizajnu. Podľa knihy C. Cowarda [22] sa tento nástroj zaraďuje medzi softvér s parametrickým modelovaním. Každá vlastnosť je definovaná súborom charakteristík zvaných parametre. Ako príklad uvádza kocku, ktorá sa skladá z premenných, napríklad pozícia stredu kocky, jej výška, šírka a hĺbka. Uvedené rozmery sú v podstate parametrami využívanými softvérovým nástrojom na popis tohto trojrozmerného objektu. Pomocou softvéru je následne možné tieto parametre uložiť a v prípade potreby kedykoľvek upravovať. Využívanie parametrického modelovania je v inžinierstve veľmi dôležité z hľadiska viacerých aspektov. Prvým z nich je spoľah na presne definované rozmery. Tie sú dôležité, ak chceme namodelovať súčiastky, ktoré do seba musia zapadať. Coward vo svojej knihe ako príklad uvádza piest v spaľovacom motore. Ten musí byť určený takými parametrami, aby sa vošiel do piestneho priestoru motora.

Na vývoj dielov robotického ramena sme si zvolili práve tento softvér z dôvodu robustnosti a flexibility pri práci. Oproti softvéru Tinkercad poskytuje oveľa viac funkcií modelovania, vyššiu kvalitu výsledného súboru (vyššie vzorkovanie na popis kriviek), profesionálnejšiu manipuláciu s 3D modelmi. Navyše disponuje aj tzv. časovou osou, čo je veľkou výhodou najmä pri iteratívnom dizajne, kedy je možné dodatočne pozmeňovať hodnoty už zadaných parametrov. Hlavné okno CAD systému Fusion 360 (Obrázok 4–1) obsahuje nasledujúce základné časti:

- pracovný priestor (z angl. *workspace*) reprezentovaný mriežkou,
- vrchný panel nástrojov,
- spodný panel s časovou osou aktuálne označeného dizajnu,
- ľavý panel s hierarchickou štruktúrou telies, nákresov a ďalších prvkov.

V pracovnom priestore používateľ využitím nástrojov vrchného panela navrhuje 3D dizajn. Základným prvkom akéhokoľvek návrhu dizajnu je nákres – môže to byť ná-



Obrázok 4–1 Hlavné okno prostredia Fusion 360

rys, pôdorys alebo bokorys, v závislosti od roviny, na ktorej je definovaný. Nákres je dvojrozmerný geometrický profil tvoriaci základ 3D dizajnu modelu. Pred začatím konštrukcie 3D objektov v rámci aktuálne vytváraného návrhu je potrebné načrtnúť základné profily nákresu, ktoré následne riadia celkové tvary parametrického výsledného telesa, prípadne jeho povrchu. Je základom každého parametrického modelovania a pri zhotovení adaptívneho nákresového profilu sa môže výrazne zjednodušiť pracovný postup a minimalizovať tak potenciálne vzniknuté problémy. Ďalším krokom je dvojrozmerný nákres expandovať do tretieho rozmeru. Realizácia prebieha vysunutím plochy profilu nákresu o požadovaný počet jednotiek do konkrétneho smeru po vybranej osi, čím vznikne trojrozmerný objekt. Takýmto postupom vieme nadizajnovať prakticky akýkoľvek ľubovoľný tvar v trojrozmernom kartézskom súradnicovom systéme. Softvér Fusion 360 poskytuje aj niektoré pokročilé funkcie a nástroje slúžiace používateľovi na efektívnejšiu manipuláciu s aktuálne navrhovaným modelom.

4.1.2 Tvorba konštrukčného návrhu

Pred samotným začatím tvorby konštrukčného návrhu robotického ramena bolo premyslieť všetky jeho technické detaily. Patrí medzi ne určenie počtu stupňov voľnosti, hrubý náčrt tvaru skeletu ramena, prípadná inšpirácia už existujúcim priemyselným ramenom, zvolenie vhodného konštrukčného materiálu, výber aktuátorov (prípadne senzoričky), aktuálna dostupnosť zvolených elektronických komponentov na trhu a podobne. Konceptná štúdia modelovania robotického ramena začína jednoduchým náčrtom. Z hľadiska lepšej predstavivosti počiatočný návrh nebol zhotovený ručne na papieri, ale využili sme jednoduchý nástroj použitý v bakalárskej práci – Autodesk Tinkercad. Na obrázku 4–2 je zobrazený počiatočný náčrt ramena (vľavo), inšpirovaný tvarom priemyselného robotického ramena Mitsubishi RV – 2SDB (vpravo). Z uvedeného porovnania je zrejmé, že inšpirácia sa zameriavala



Obrázok 4–2 Porovnanie počiatočného návrhu ramena a priemyselného ramena Mitsubishi

najmä na celkovú geometriu a dizajnovú stránku ramena. To zahŕňa usporiadanie kĺbov, tvary a dĺžky medzikĺbových prepojení a v neposlednom rade aj ukrytie všetkej kabeľáže do konštrukcie robota. Bolo tiež potrebné zvážiť, aký typ aktuátorov bude použitý s čím súvisia aj ich technické parametre. Analýzou ramena Mitsubishi sme zistili, že na druhý a tretí stupeň voľnosti je vyvíjaný vysoký mechanický stres, v dôsledku čoho musia byť zvolené také aktuátory, ktoré umožňujú silové pôsobenie zvládnuť. Pri návrhu je nutné myslieť aj na napájanie akčných členov a senzorov.

Ukrytie kabeláže do tela ramena vytvára potrebu zaviesť určitú konvenciu napájania elektrickou energiou a vedenia signálov. Riešenie spočívalo vo výbere štandardne zaužívaných konektorov a príslušných káblov zvolených podľa určitých špecifikácií.

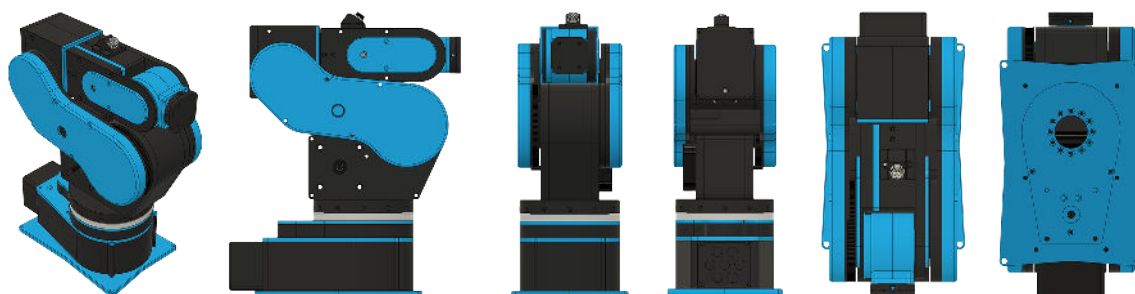
Po analýze sa určili konkrétne prvky a parametre robotického ramena. Ako pohonné jednotky kĺbov sú zvolené krokové motory pre ich vysokú presnosť a spoľahlivosť. Pri správnom sprevodovaní poskytujú aj dostatočný ťah. Sprevodované sú motory prvého až piateho kĺbu s využitím ozubených remeňov. Týmto krokom sa zvýši krútiaci moment kĺbov a rameno je tým schopné spoľahlivého ľubovoľného pohybu. Os šiesteho stupňa voľnosti je priamo prepojená s hriadelom krokového motora. Na zaručenie plynulého otáčania kĺbov každý kĺb obsahuje ložisko, prípadne viacero ložísk. Podobne, ako pri bakalárskej práci sú jednotlivé navrhnuté diely konštrukcie ramena vytlačené pomocou technológie 3D tlače metódou fúzej dekompozície [2]. Použitým materiálom je kyselina polymliečna (PLA). Výhodou tohto materiálu je jednoduchá manipulácia, cenová dostupnosť a relevantná pevnosť. Jednotlivé výtlačky a aktuátory sú do funkčného celku pospájané štandardným spojovacím materiálom normy DIN.

Priemyselné robotické ramená využívajú na zisťovanie polohy kĺbov senzorku umožňujúcu poskytovať spätnú väzbu. Údaje o polohe si pomocou rotačných enkóderov či iných podobných senzorických prostriedkov odosielajú do riadiacej jednotky, kde si ich ukladajú do pamäte. To im umožňuje znižovať chybovosť pri kinematických výpočtoch a zároveň zvyšovať svoju presnosť pri opakovaných pohyboch. Navyše nie je nutné pred každým spustením robota centrovat všetky stupne voľnosti. Z dôvodu uľahčenia tejto úlohy sme sa ale rozhodli tento druh senzorky do ramena neimplementovať nielen z finančného hľadiska, ale aj kvôli šetreniu priestorom v konštrukcii. Pri návrhu alternatívneho riešenia, ako zistiť polohy jednotlivých kĺbov, sme sa inšpirovali 3D tlačiarňou, ktorá disponuje pomerne jednoduchým spínačovým systémom. Pre každú riadenú os je k dispozícii jeden dotykový mikrospínač schopný binárnej spätnej väzby. Pri vybudení logickej jednotky si program uloží aktuálne nadobudnutú polohu a označí ju ako východziu. Pred spustením robotického ramena

sa pre každú zo šiestich osí vykoná kalibračná rutina, ktorá zabezpečí nastavenie kĺbov do počiatočného kinematického stavu (Obrázok 4–4). Výsledkom kalibrácie je vycentrované robotické rameno s kĺbami nastavenými na uhol 0° , čím je pripravené na prevádzku.

Proces tvorby návrhu komponentov pre robotické rameno sme začínali návrhom základne. Na nej spočíva celá váha ramena. Navyše otáča celou zvyšnou vrchnou časťou, nech už je rameno v akejkoľvek kinematickej polohe. Najspodnejšia časť ramena preto musí vykazovať dostatočnú pevnosť, nosnosť a robustnosť. Os otáčania prvého rotačného kĺbu je umiestnená viac v prednej časti a obsahuje dvojicu ložísk. Ložiská zabezpečujú hladké a plynulé otáčanie kĺbu okolo osi z , čo do určitej miery ovplyvňuje aj konečnú presnosť koncového bodu. V zadnej časti základne sú otvory na montáž štandardných priemyselných konektorov s označením GX-12. Slúžia ako vstupno-výstupné vývody pre napájanie, riadenie krokových motorov a tiež na prenos signálov zo spínačov. Pohonnými jednotkami ramena sú krokové motory z typu označovaného ako National Electrical Manufacturers Association (NEMA), známy najmä vďaka širokému výskytu v 3D tlačiarňach. Na trhu existuje viacero druhov v závislosti od technických špecifikácií motora ako je napríklad ťah, veľkosť, polarita a podobne. V základni ramena je umiestnený krokový motor NEMA 23, ktorý v spojení s ozubeným remeňom zabezpečuje dostatočný ťah na otáčanie zvyšnou časťou robota. Nad základňou sa nachádza druhý stupeň voľnosti s krokovým motorom rovnakého druhu ako v základni, nakoľko na tento kĺb je vyvíjaný najväčší moment sily. V strednej časti robotického ramena je umiestnený aj aktuátor pre tretí stupeň voľnosti s označením NEMA 17. Líši sa od predošlých dvoch aktuátorov svojou veľkosťou a disponuje nižším ťahom, no pre potreby aktuácie tohto kĺbu parametricky postačuje. Vrchná časť robota je do značnej miery inšpirovaná skutočnou predlohou. Má štvorcový prierez a zahŕňa v sebe posledné tri kĺby, ktoré zdieľajú rovnaký druh použitého motora – NEMA 14. Pridruženou súčasťou každého zo šiestice kĺbov je vyššie spomenutý mikrosplínač schopný detekcie pevného absolútneho uhlu natočenia, od ktorého si dokáže program riadenia odvodiť ľubo-

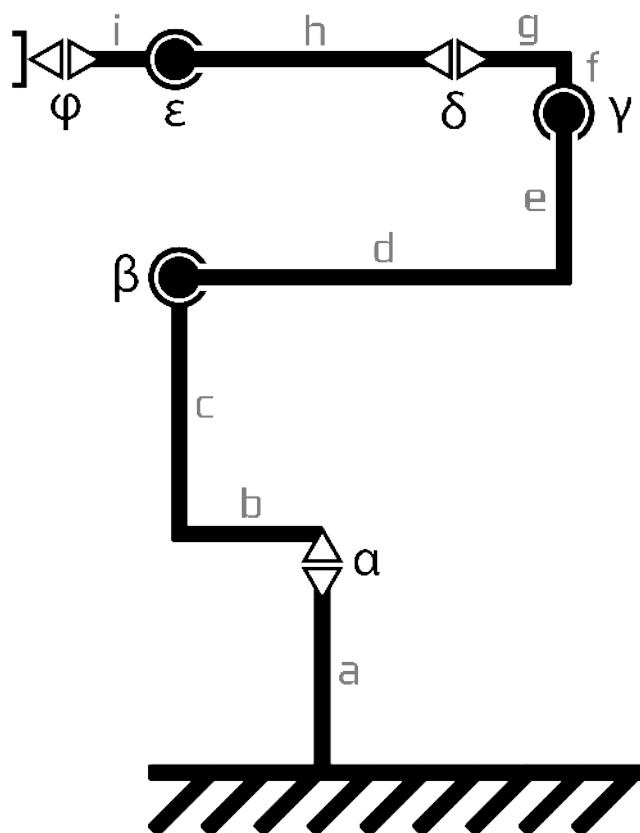
voľnú pozíciu kĺbu z povoleného rozsahu. Na koncovom bode ramena sa nachádza univerzálny nástavec pre montáž nástroja. Môže ním byť úchopný mechanizmus alebo elektromagnet. V diplomovej práci sú však na manipuláciu s objektmi využité spomínané úchopné čeluste. Na napájanie a prenos signálov z nástroja ramena slúži špeciálny dedikovaný konektor GX-12. Výsledný návrh konštrukcie robotického ramena je zobrazený na obrázku 4–3, kde je okrem globálneho pohľadu na návrh vyobrazený aj bokorys, nárys a pôdorys modelu. Rovnako, ako pri bakalárskej práci



Obrázok 4–3 Výsledný 3D návrh modelu robotického ramena

bola z estetického hľadiska zvolená dvojfarebná, tentokrát modro-čierna kombinácia. Po zhotovení celkového návrhu robotického ramena a prejdení k jeho realizácii bolo finálnym krokom z hľadiska optimalizácie skontrolovať všetky kritické diely a súčiastky, aby sa zamedzilo vzniku možných problémov pri fabrikácii.

Vzhľadom k tomu, že na riadenie robotických ramien je nutné poznať ich kinematickú štruktúru, resp. kinematický reťazec, vznikla potreba zadefinovať všetky rotačné a translačné prvky konštrukcie ramena. Prvým krokom bolo pre zľahčenie meraní hodnôt translácií vykonať prierezovú analýzu (z angl. *cross-section analysis*), čo je jeden z nástrojov poskytovaných softvérom Fusion 360. Týmto úkonom sa rozdelilo rameno pozdĺž vertikálnej roviny na polovicu a následne sa odčítali všetky translačné zložky s poznačením pozícií bodov pre rotácie. Kompletná kinematická štruktúra sa nachádza na obrázku 4–4 a pod ním sú v tabuľke 4–1 uvedené hodnoty symbolov vyznačených v obrázku písmenami. Sčítaním translačných zložiek sa zistilo, že robotické rameno natiahnuté do vzpriamenej polohy je vysoké približne 600mm (bez namontovaného nástroja). Váha robota je približne 7600g .

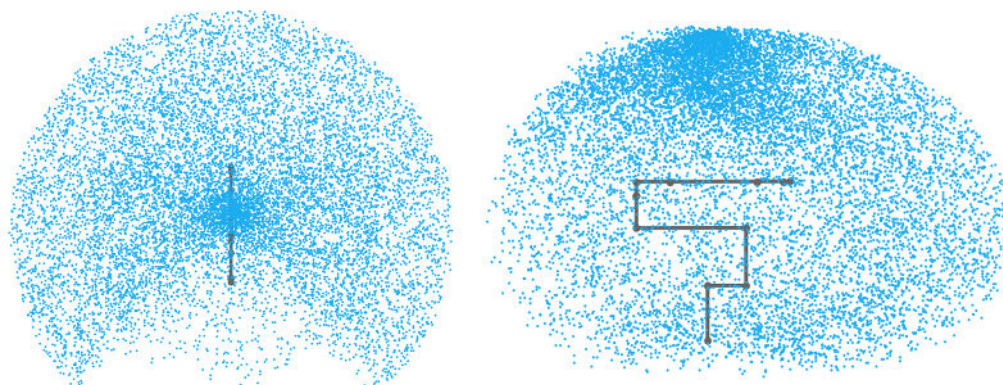


Obrázok 4–4 Schematický náčrt kinematickej štruktúry

Translačná časť	Hodnota [mm]	Rotačná časť	Rozsah [°]
a	98.50	α	$\langle -135; 135 \rangle$
b	52.00	β	$\langle 0; 180 \rangle$
c	104.00	γ	$\langle 0; 150 \rangle$
d	148.30	δ	$\langle -90; 90 \rangle$
e	58.30	ε	$\langle -120; 120 \rangle$
f	25.00	φ	$\langle -180; 180 \rangle$
g	45.00	—	—
h	117.50	—	—
i	36.00	—	—

Tabuľka 4–1 Vysvetlenie symbolov ku kinematickej štruktúre

Konštrukčné vlastnosti ramena postačujú na riešenie zadaných úloh sklada-
nia. K pojmu kinematickej štruktúry sa viaže pojem pracovného priestoru manipu-
látoru. Popisuje ohraničenú množinu bodov v kartézskom priestore, zväčša vyjadre-
ných súradnicami (x, y, z) , ktoré popisujú všetky možné stavy koncového efektora ki-
nematického reťazca. Pri priemyselných robotických ramenách je udávaný výrobcom
priamo v dokumentácii presným geometrickým vyjadrením vychádzajúcim z kĺbo-
vého usporiadania. Zjednodušene povedané opisuje, na aké miesta sa koncový bod
ramena môže dostať. Pre uľahčenie určenia a vizualizácie pracovného priestoru na-
vrhnutého ramena sme vygenerovali pomerne rozsiahlu množinu bodov. Každý bod



Obrázok 4–5 Bodová vizualizácia pracovného priestoru ramena

je získaný výpočtom priamej kinematiky uvedenej kinematickej štruktúry manipu-
látoru na základe stochasticky generovaných polôh individuálnych kĺbov. Výsledkom
je mračno bodov zvýraznené modrou farbou (Obrázok 4–5) okolo štruktúry vyzna-
čenej tmavosivou farbou. Body približne kopírujú ohraničenia kinematického modelu
robotického ramena.

4.1.3 Realizácia konštrukcie ramena

Významnou etapou v procese realizácie konštrukcie modelu robotického ramena je
príprava 3D objektov. V počítači sa kódujú špecializovanými súborovými formátmi,
ako sú napríklad Stereolithography (STL), Wavefront Object File (OBJ) či novší 3D

Manufacturing Format (3MF). Každý z nich sa od seba líšia svojimi vlastnosťami. V súčasnosti najpoužívanejším spomedzi vymenovaných formátov je práve stereolitografický formát. Trojrozmerný objekt je reprezentovaný trojuholníkovou mriežkou a pri binárnom kódovaní súboru do značnej miery šetrí pamäťovým priestorom. Zároveň svoju popularitu nadobudol práve príchodom 3D tlače. 3D tlačiarne sú komerčne dostupné a tak má k tlači prístup už aj úplný začiatok v tejto problematike. STL formát je veľmi vhodným kandidátom pre interpretáciu objektu, ktorý chce používateľ fyzicky zhotoviť. Súborový formát podporuje aj CAD softvér Fusion 360, v ktorom sme celý návrh ako aj všetky ostatné podporné súčiastky dizajnovali. Exportovaním jednotlivých dielov z programu nám umožní ďalej s týmito objektami pracovať. Predspracovanie súborov, aby boli zrozumiteľné pre 3D tlačiareň, prebieha pomocou tzv. *slicing* softvéru. Detaily problematiky boli bližšie rozpísané v bakalárskej práci [2]. Momentálne najrozšírenejšia technológia 3D tlače na trhu je fúzna dekompozícia (FDM – Fusion Decomposition Modelling). Je to metóda realizácie 3D modelu postupne vrstvu po vrstve, kde tlačová hlava premieňa tuhý materiál (nazývaný filament) na roztavenú hmotu. Hmota následne tuhne, čím vytvorí vrstvu. Séria vrstiev naskladaných na seba vo vertikálnej osi utvára celý objekt. Pre potreby prípravy vymodelovaných komponentov obsahuje *slicer* viacero nastavení tlače. Patrí medzi ne teplota trysky, teplota výhrevnej podložky, tvar a hustota výplne, výška vrstvy (záleží aj od priemeru trysky), rýchlosť tlače, parametre chladenia, hrúbka stien objektu a rôzne iné atribúty. Pre naše potreby vyhotovenia jednotlivých dielov ramena sme použili tieto nastavenia:

- **Výška vrstvy** – 0.2 mm, štandardná odporúčaná hodnota, zabezpečuje dobrú pevnosť finálneho výtlačku pri relatívne rýchlej tlači,
- **Teplota tlače** – 200 °C / 60 °C, prvá hodnota determinuje teplotu trysky, druhou hodnotou je teplota podložky, na ktorú je objekt tlačný,
- **Výplň a hustota výplne** – diagonálna mriežková štruktúra s 60% nastavením hustoty,

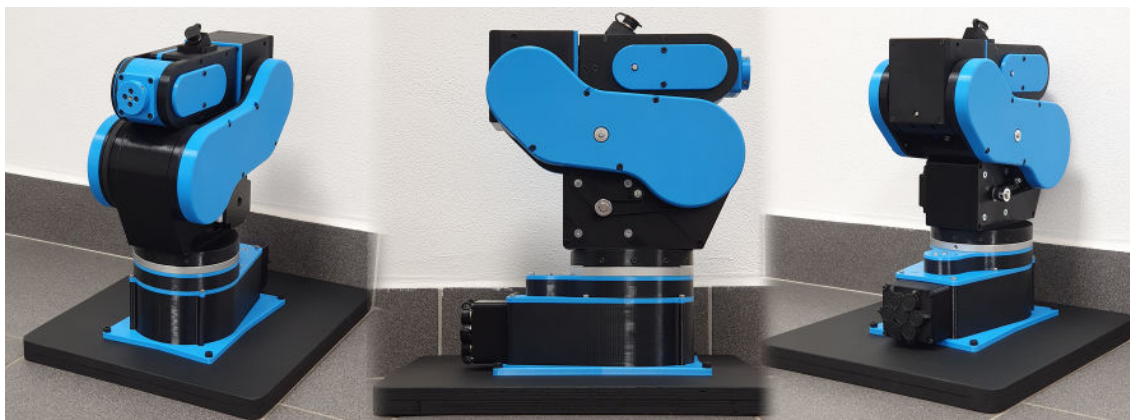
- **Rýchlosť tlače** – 60 mm/s, štandardne používaná hodnota.

Bližšie podrobnosti sú uvedené v dokumentácii záverečnej práce. Po nastavení atribútov v rámci softvéru a následnom zobrazení náhľadu, ako bude hotový výtlačok vyzeráť sa mohla začať fáza tlačenia. Softvér vygeneroval špeciálnu sadu inštrukcií pre 3D tlačiareň zvanú *geometric code* (skrátene *g-code*), ktorú je schopná interpretovať ako pohyby tlačovou hlavou. Vzhľadom k úspešnému použitiu materiálu PLA pri tlači súčiastok pre robota v bakalárskej práci sa neuvažovalo nad zmenou.

Po vytlačení kompletnej sady dielov pre robotické rameno sa pristúpilo k spájaniu do segmentov a tie následne do finálneho produktu. Výtlačky boli spájané štandardným spojovacím materiálom s označením DIN 912 a DIN 934. Do konkrétnych častí na vopred určené miesta tela konštrukcie boli namontované krokové motory a spínače. Vodiče vedúce od elektronických komponentov boli uložené do vnútornej dutiny ramena, aby nedochádzalo k ich nadmernému namáhaniu externými faktormi alebo fyzickému poškodeniu popri riadení kĺbovej sústavy. V rámci montáže sme robotické rameno postupne zhotovovali v troch fázach:

1. **základňa** s prvým stupňom voľnosti,
2. **stredná časť** s druhým a tretím stupňom voľnosti,
3. **vrchná časť** so štvrtým až šiestym stupňom voľnosti.

Zložením a otestovaním každého segmentu po funkčnej a pevnostnej stránke sa pospájali do finálneho robotického ramena (Obrázok 4–6). Rameno spĺňa navrhované požiadavky, poskytuje dostatočnú robustnosť a precíznosť na zvládnutie zadanej úlohy. Poslednou etapou pri realizácii fyzického modelu robota bolo vybrať vhodný nástroj koncového efektora. Z charakteru zadanej úlohy vyplýva, že na presun a manipuláciu s objektmi je nutné zabezpečiť určitý druh úchopného mechanizmu. Spoločnosť BCN3D v roku 2016 vydala open-source návrh 3D tlačeného robotického ramena s názvom MOVEO [23]. Súčasťou ich návrhu boli aj robotické servom poháňané úchopné čeluste s maximálnou šírkou úchopu 70 mm. V záujme urýchlenia



Obrázok 4–6 Finálna fyzická podoba modelu robotického ramena

procesu finalizácie a s prihliadnutím na fakt, že BCN3D MOVEO projekt nepodlieha licencií sme tento úchopný mechanizmus prevzali a integrovali do nášho robotického ramena. Výsledok je znázornený na obrázku 4–7 spoločne s konektorom vo vrchnej časti robota slúžiaceho na napojenie servopohonu. Použitý servomotor s označením MG995 je adekvátny na plnenie svojej úlohy vďaka dostatočnej sile úchopu a pohodlnému programovému riadeniu. Navyše je možné nastavovať rýchlosť zovretia



Obrázok 4–7 Robotické rameno s namontovaným úchopným mechanizmom

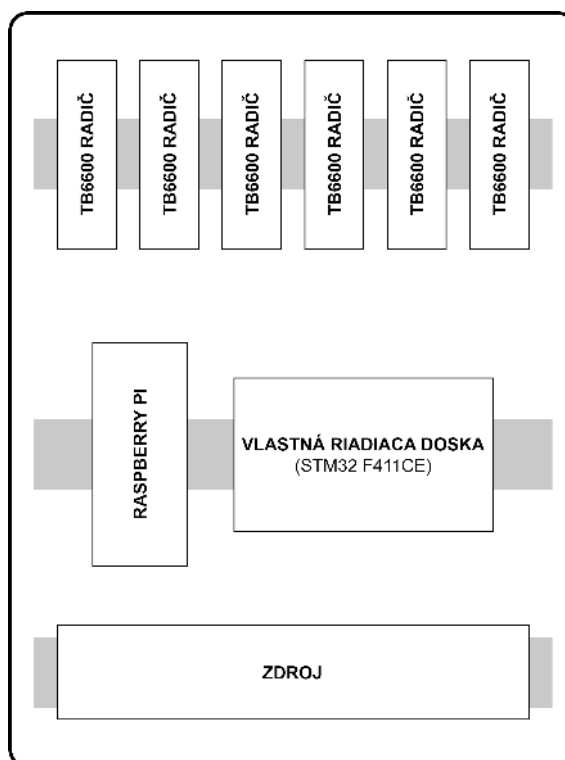
čelustí, keďže pracuje na pulzno-šírkovej modulácii (PWM) signálu generovaného riadiacou jednotkou. Koncovka dodávaná výrobcom motora bola nahradená priemyselným konektorom GX-12 za účelom štandardizácie.

4.2 Návrh a realizácia riadiacej jednotky robotického ramena

Značná časť robotických ramien ponúkaných na trhu má svoje riadenie oddelené od samotnej konštrukcie. Prakticky to znamená, že výpočtové jednotky a ostatné elektronické komponenty ako radiče motorov alebo zdroj sú uložené mimo tela robota. Ako príklad si uvedieme produkt od spoločnosti KUKA AG. Je ním produkt KUKA KR C4 [24], čo je riadiaci systém pre priemyselné robotické ramená vyrábané touto spoločnosťou. Všetky riadiace členy sú na uloženie na jednom mieste a sprostredkujú prenos dôležitých elektrických signálov medzi riadením a ramenom.

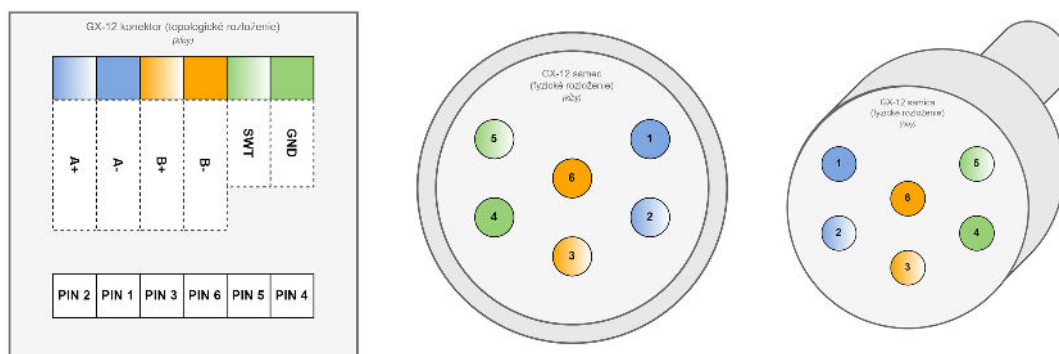
4.2.1 Návrh architektúry riadiacej jednotky ramena

Myšlienku centralizácie sme do určitého stupňa využili už aj v bakalárskej práci na vytvorenie riadiacej architektúry pre robotické rameno, avšak predošlé riešenie nebolo vo viacerých ohľadoch premyslené a ani nedodržiavalo priemyselné štandardy. Pri tvorbe nového návrhu ramena sme sa preto inšpirovali vyššie spomenutým prístupom. Ako ukazuje schematický náčrt (Obrázok 4–8), enkapsuláciou celého riadiaceho systému do jedného funkčného bloku – riadiacej skrinky sú všetky podstatné riadiace komponenty na jednom mieste. Vnútri skrinky sa nachádza zdroj elektrickej energie, ovládače pre krokové motory, plošný spoj s jednočipovým mikropočítačom a vstupno-výstupnými konektormi a jednodoskový počítač Raspberry Pi. Všetky vymenované komponenty sú uchytené na štandardných priemyselných lištách typu DIN (na obrázku 4–8 vyznačené svetlosivou farbou). Každý z komponentov má špeciálne navrhnutý držiak na túto lištu, aby bola zabezpečená rýchla montáž alebo demontáž v prípade údržby, pokiaľ to bude nutné. Na bočnej stene riadiacej skrinky sú umiestnené vstupno-výstupné konektory pre kabeláž smerujúcu ku robotickému ramenu, hlavný prívodný kábel od zdroja napájania a tiež sa tu nachádza aj spínačový systém. Veľmi dôležitou súčasťou návrhu architektúry riadenia je okrem iného aj zavedenie vodičovej konvencie. Analýzou sa zistilo, aké požiadavky elektrické vodiče



Obrázok 4 – 8 Konceptia riadiacej skrinky

pre napájanie ramena potrebujú splňať na správne fungovanie s cieľom poskytnúť dostatočnú prúdovú záťaž. Riadenie ktoréhokoľvek kľbu vyžaduje šesticu káblov nevyhnutných na obsluhu daného stupňa voľnosti. Štyri z nich sú určené pre napájanie cievok krokového motora elektrickou energiou. Zvyšné dva vodiče zasa vedú k mikrosopínačovému snímaču. Nakoľko robotické rameno obsahuje šesť kľbov je napájanie každého z nich riešené šesť-žilovým vodičom. Vodič je ukončený z oboch strán šesťpinovým priemyselným konektorom GX-12. Schematický obrázok 4–9 znázorňuje topologické i fyzické rozloženie signálov vnútri vodiča. Symboly $A+$, $A-$, $B+$, $B-$ označujú cievky bipolárneho krokového motora NEMA a symboly SWT a GND vyjadrujú elektrické dráhy mikrosopínača. Ako médium bol zvolený Foiled Twisted Pair (FTP) kábel s Cat 5e certifikáciou. Priemer medenej žily uvádzaný ako americká jednotka American Wire Gauge (AWG) s hodnotou 24 AWG postačuje na bezproblémové vedenie elektrickej energie do krokových motorov. Pred finálnym nasadením

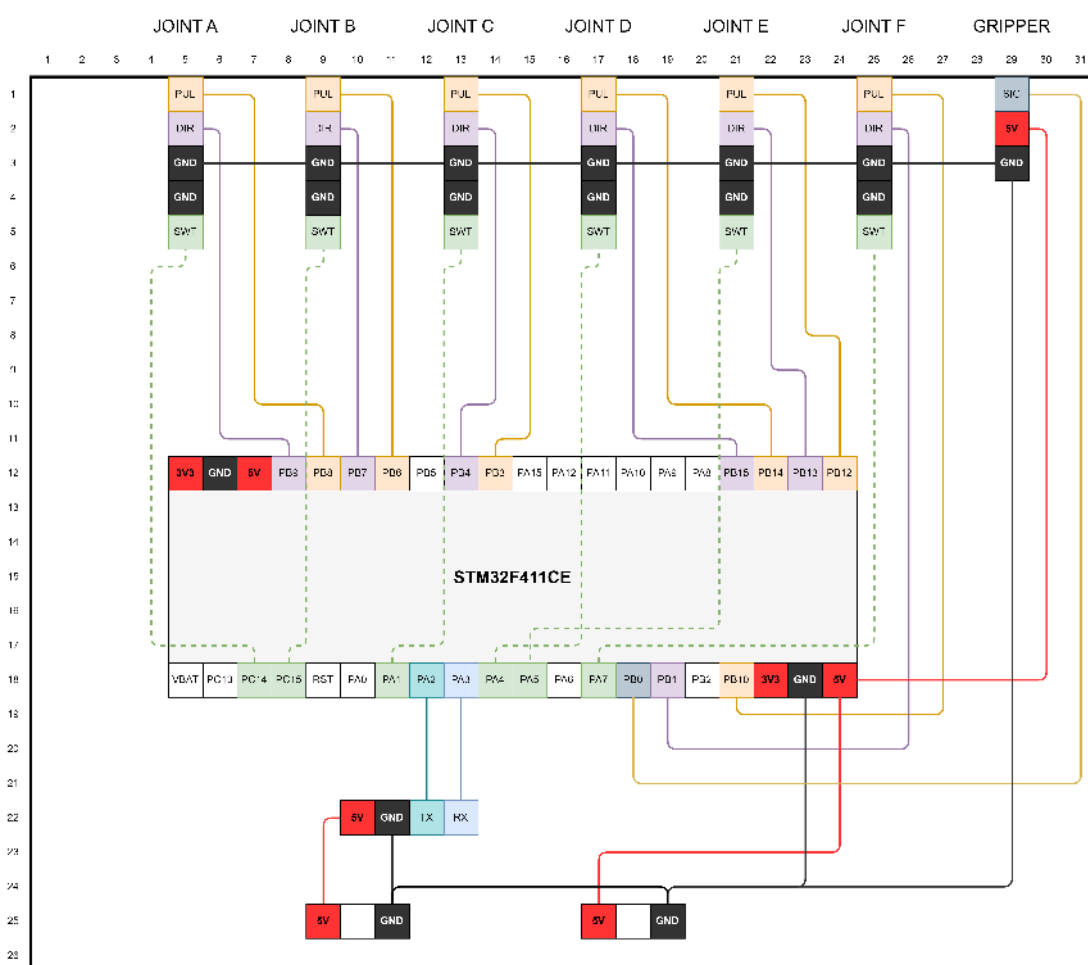


Obrázok 4–9 Schematický popis rozloženia signálov vo vodiči

sme dôkladne otestovali spôsobilosť tohto typu kabeláže. V rámci uvedenej koncepcnej schémy (Obrázok 4–8) sa v jej hornej časti nachádza sada radičov pre krokové motory. Mikrokrokový radič s označením TB6600 je postavený na integrovanom obvode spoločnosti Toshiba a možno ho použiť na pohon dvojfázových bipolárnych krokových motorov. Vzhľadom k tomu, že každý jeden kĺb je aktuovaný jedným krokovým motorom, je na pohon celého robotického ramena vyžadovaných šesť radičov. Maximálny prúdový odber radiča je 4 A čo zaisťuje bezproblémový chod pre väčšinu krokových motorov, vrátane tých v našom robotickom ramene. Radič má navyše zabudovaných niekoľko bezpečnostných funkcií, ako je protiprúdová ochrana, vypnutie pri vzniku podpätia či ochrana proti prehriatiu. Radič TB6600 obsahuje aj prepínačový systém pre tzv. *microstepping* mód. Vyjadruje akúsi mieru precíznosti jedného kroku motora, pretože rozdeľuje krok na menšie časti.

Na riadenie krokových radičov, snímanie senzorických vstupov a riadenie úchopného mechanizmu ramena bolo potrebné zabezpečiť výber vhodnej výpočtovej jednotky spôsobilej pre tieto úkony. Jednotka musí poskytovať dostatočný výpočtový výkon pre prácu obsluhu šiestich kĺbov, nakoľko riadenie sa realizuje paralelne. Dostatočné výpočetné kapacity na splnenie definovaných požiadaviek má jednočipový mikropočítač od spoločnosti ST Microelectronics, konkrétne typ STM32F411CEU6. Disponuje 512 KB flash pamäťou, 128 KB SRAM a podporuje množstvo zberníc pre komunikáciu – Inter-Integrated Circuit (I2C), Universal Asynchronous Receiver-

Transmitter (UART), Serial Peripheral Interface (SPI). Ďalej poskytuje 10 analógových vstupov, 32 digitálnych vstupno-výstupných signálov, pričom 25 z nich podporuje PWM. Frekvencia procesora dosahuje 100 MHz, čo pre naše účely plne postačuje. Na báze tohto čipu existuje vývojová doska s názvom *Black Pill*, ktorá v sebe integruje USB port a vstupno-výstupné signály sú vyvedené ako piny na okrajoch plošného spoja dosky pre pohodlnejšiu prácu s mikročipom. Pripojenie všet-



Obrázok 4 – 10 Schéma vlastnej riadiacej dosky

kých periférií na vývojovú dosku si vyžadovalo aj návrh vlastnej prototypovacej dosky schopnej pripojenia všetkých šiestich radičov krokových motorov a servomotora úchopného mechanizmu. Schematický náčrt dosky sa nachádza na obrázku 4–10. Na doske sú umiestnené vstupy pre napájanie jednočipového mikropočítača

a jednodoskového počítača Raspberry Pi linkou +5V zo zdroja. V spodnej časti prototypovacej dosky je vyvedená štvorica pinov sprostredkujúca sériovú komunikáciu. Dvojica RX a TX slúži na prenos dát, napájanie pre Raspberry Pi zaobstarávajú výstupy s označením *GND* a +5V. Priemyselný spínaný zdroj s výstupným napätím +12V a výkonom 360W poskytuje celej riadiacej sústave dostatočné napájanie.

4.2.2 Realizácia riadiacej jednotky ramena

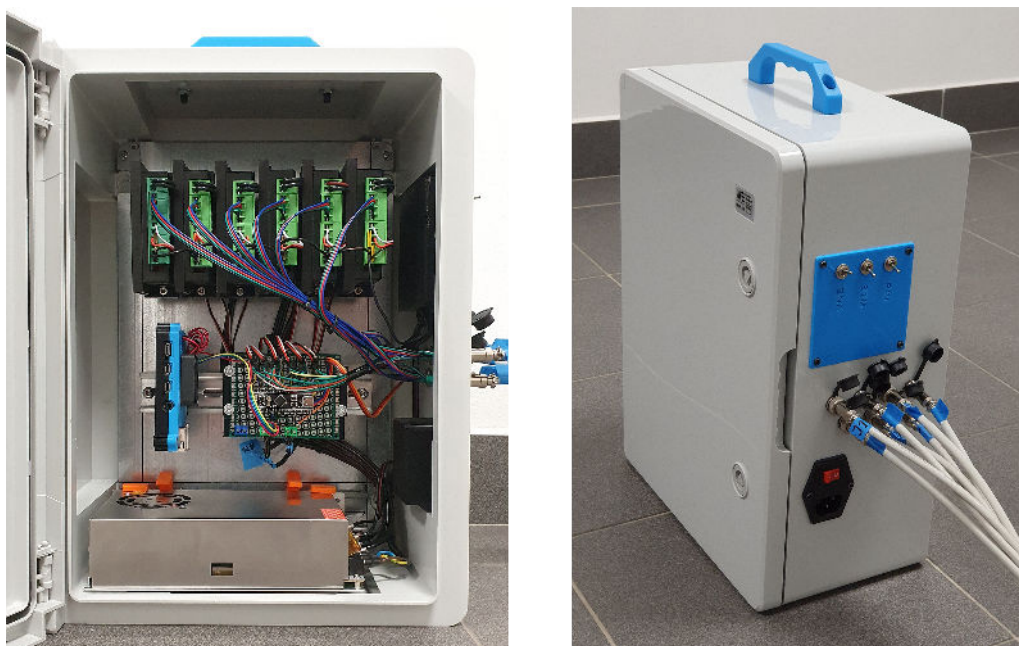
Po zavedení architektúry riadenia sme mohli prejsť k jej realizácii. Celé riadenie je enkapsulované do skrinkového rozvádzača vyrobeného z pevného plastu s rozmermi $5mm \times 5mm \times 5mm$. Na zadnej stene skrinky je umiestnený pozinkovaný plech s uchytenými DIN lištami. Na lištách sú špeciálnymi držiakmi vytlačenými na 3D tlačiarňi nainštalované všetky elektronické komponenty. Držiaky sú výhodou najmä pri výmene alebo údržbe ľubovoľného komponentu.

Vrchnú časť skrinky zaberajú radiče pre krokové motory o počte šesť kusov. V strednej časti sa nachádza počítač Raspberry Pi 4 a vlastná doska riadenia s mikroprocesorom STM32. Hlavný elektrický zdroj je situovaný na dne skrinky a taktiež je prichytený o DIN lištu držiakmi. Na vrchu skrinky je upevnený praktický držiak pre jej pohodlný presun z miesta na miesto. Na pravej bočnej stene je sedem GX-12 konektorov. Šesť z nich slúži na pripojenie kĺbov ramena (krokové motory a spínače) a jeden na pripojenie koncového nástroja. Prívod striedavého prúdu 230V je vedený do skrinky štandardnou IEC320 C14 zásuvkou s prepínačom a poistkou. Slúži ako hlavný vypínač riadenia robotického ramena. Napájanie elektronických komponentov vnútri skrinky je vyriešené spínačovým systémom umiestneným v hornej časti bočnej steny (Obrázok 4–11 vpravo). Pozostáva z trojice elektrických spínačov, pričom každý z nich zastáva určitú funkcionálnosť:

- spínač **ENA** – pri zopnutí deaktivuje všetky radiče krokových motorov, čo má za následok okamžité prerušenie prívodu elektrickej energie do cievok motorov; slúži ako brzdiaci mechanizmus kĺbov ramena,

- spínač **STM** – ovláda elektrický prívod ku hlavnej riadiacej doske, na ktorej sa nachádza mikroprocesor STM32 s obslužným programom riadenia,
- spínač **RPI** – ovláda elektrický prívod ku jednodoskovému počítaču Raspberry Pi.

Spínačový systém izoluje funkčné celky riadenia zapínaním, respektívne vypínaním konkrétnych členov riadenia. Z hľadiska bezpečnosti je výhodné disponovať spínačom pre odstavenie motorov, aby sa pri prípadnej kolízii ihneď odstavil pohon kĺbov a nedošlo k poškodeniu konštrukcie a elektroniky robota. Výsledná podoba riadiacej skrinky pre robotické rameno sa nachádza na obrázku 4–11. Ľavá časť obrázka zobrazuje fyzické usporiadanie elektronických komponentov vnútri skrinky. Na pravej strane obrázka je v prostrednej časti spínačový systém, pod ktorým sú do príslušných konektorov pozapájané vodiče vedúce od robotického ramena.



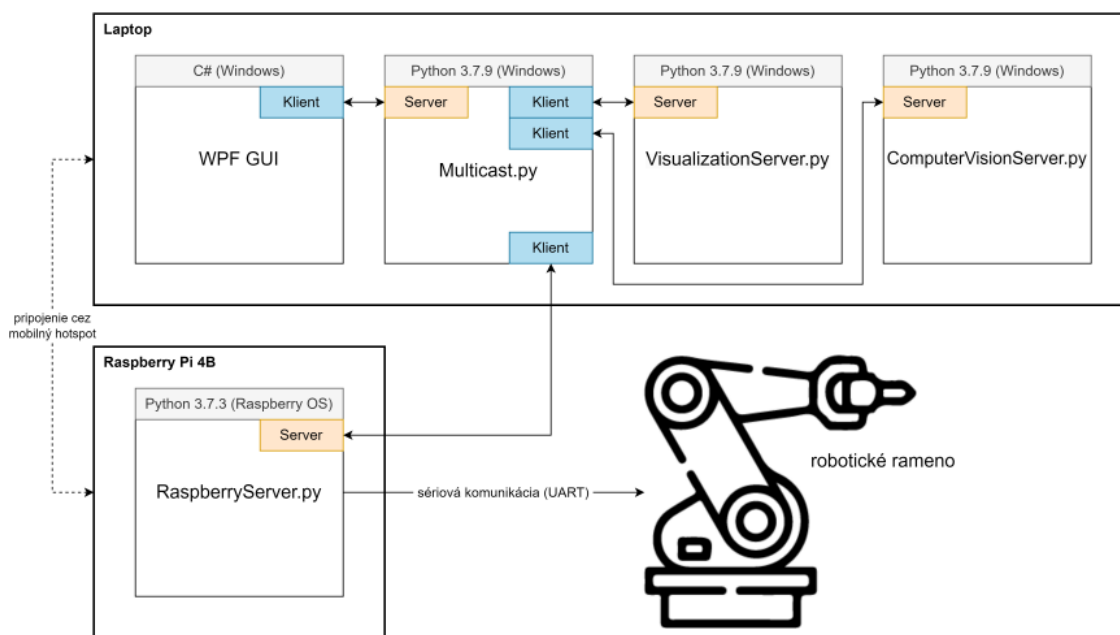
Obrázok 4–11 Finálna podoba riadiacej skrinky robotického ramena

4.3 Architektúra riadenia

Architektúra riadenia vychádza z konceptu použitého v bakalárskej práci. Zatiaľ čo predošlá koncepcia riadenia bola postavená na lokálnych výpočtoch priamo na jednodoskovom počítači Raspberry Pi je aktuálna architektúra založená na decentralizovanom systéme riadenia. Tento systém vo všeobecnosti pozostáva z menších celkov, kde sú signály privádzané do počítačov, ktoré sú v blízkosti technológie a tie zabezpečujú lokálne riadenie. Sú vzájomne prepojené a komunikujú s nadradenými počítačmi, ktoré koordinujú ich činnosť.

4.3.1 Návrh architektúry riadenia

Sumarizačný náčrt architektúry riadenia je zobrazený na obrázku 4–12 so symbolickým znázornením komunikačných tokov prostredníctvom šípok. Bloková štruktúra rozdelená do viacerých častí vyjadruje konkrétne topologické prepojenia programov riadenia a ich súvislosti. Dáta smerujú z grafického rozhrania spúšťaného na operačnom systéme Windows do aplikácie Multicast. Bloky vrchného segmentu označe-



Obrázok 4–12 Schematický náčrt architektúry riadenia

ného na obrázku ako *Laptop* sú navzájom prepojené cez tzv. localhost so všeobecne definovanou IP adresou 127.0.0.1, pričom každá aplikácia má navyše definovaný aj port zaistujúci identifikáciu aplikácie. V tabuľke 4–2 je spísaný zoznam jednotlivých IP adries a ich príslušných portov. Zo schémy je zrejmé, že grafické rozhranie sa ako klientska aplikácia pripája na server *Multicast* (posledný riadok v tabuľke) a so zvyšnými aplikáciami rozhranie komunikuje práve cez tohto sprostredkovateľa. Prenášané dáta sú zakódované vo formáte JavaScript Object Notation (JSON), ktorý sme vybrali z hľadiska jednoduchšej manipulácie v rôznych programovacích jazykoch. Na základe IP adries pripojených serverov sa program rozhoduje, s ktorým z nich

Aplikácia	IP adresa	Port
VisualizationServer	127.0.0.1	27001
ComputerVisionServer	127.0.0.1	27002
RaspberryServer	192.168.137.94	27003
Multicast	127.0.0.1	27000

Tabuľka 4–2 Zoznam konfiguračných údajov pre jednotlivé aplikácie

bude komunikovať a odosielať údaje v podobe serializovaných JSON objektov. Veľkou výhodou takéhoto prístupu je schopnosť výmeny dát medzi viacerými spustenými programami na počítači bez nutnosti využitia pokročilejších prístupov akými sú napríklad semafor alebo zdieľaná pamäť. Zároveň tento spôsob umožňuje obojsmernú výmenu dát medzi jednodoskovým počítačom Raspberry Pi a notebookom, pretože sú tieto zariadenia pripojené k rovnakej počítačovej sieti prostredníctvom mobilného prístupového bodu.

4.3.2 Realizácia architektúry riadenia

Riadenie je rozdelené na časť s výpočtami odohrávajúcimi sa na stolnom počítači a časť s riadiacimi procedúrami spúšťanými na lokálnych zariadeniach vnútri riadiacej skrinky. Ako stolný počítač s operačným systémom Windows sa používa notebook. Komunikácia medzi notebookom a jednodoskovým počítačom Raspberry Pi

je realizovaná klient-server architektúrou. Pre zabezpečenie spoľahlivej komunikácie medzi klientskou aplikáciou a serverom je použitý protokol TCP, ktorý zaručuje doručenie údajov v správnom poradí a bez strát. Hlavné jadro riadiaceho programu pozostáva z piatich aplikácií:

1. aplikácia *WPF GUI* je grafické používateľské rozhranie naprogramované v jazyku C# s použitím technológie *Windows Presentation Foundation* (WPF),
2. aplikácia *VisualizationServer* zaobstaráva vizualizáciu robotického ramena realizovanú knižnicou *Open3D* [25] v jazyku Python,
3. aplikácia *ComputerVisionServer* má na starosti získavanie hĺbkovej mapy zo senzoru Kinect, jej predspracovanie, detekciu a identifikáciu najvyššieho objektu s následnou interpretáciou objektu do štvorhodnotovej dátovej štruktúry,
4. aplikácia *RasperryServer* je spustená na jednodoskovom počítači Raspberry Pi vnútri riadiacej skrinky, zabezpečuje príjem dát z notebooku a prenos dát cez sériovú linku do mikroprocesora STM32,
5. aplikácia *Multicast* predstavuje komunikačného mediátora medzi grafickým rozhraním a ostatnými uvedenými serverovými aplikáciami.

Implementácia počítačového videnia je popísaná v nasledujúcej kapitole 5. Každá z uvedených aplikácií zastáva určitú úlohu v procese riadenia. Používateľské rozhranie a vizualizácia sú bližšie rozoberané v kapitole 6. Aplikácia *Multicast* zabezpečuje prenos údajov medzi používateľským rozhraním a ostatnými aplikáciami. Kinematické výpočty pre robotické rameno sa odohrávajú na notebooku a vychádzajú z užívateľských akcií vykonaných v grafickom používateľskom rozhraní spustenom na operačnom systéme Windows. Rovnako je na tomto počítači spustená aj vizualizácia ramena, ktorá predstavuje virtuálne dvojča robota. Keďže hlavným cieľom záverečnej práce je premiestňovanie detegovaných objektov na odkladaciu plochu s následným poskladaním jednoduchej trojkomponentovej súčiastky, riadiaca infraštruktúra musí byť schopná reagovať na vizuálne vnemy. Senzor Kinect umiestnený

na stojane poskytuje obrazové dáta. Algoritmus počítačového videnia zistí súradnice a typ najvyššieho objektu nájdeného v pracovnom priestore. Na základe týchto dát program vygeneruje a vykoná sekvenciu príkazov pre uchopenie a presun objektu na špecifické súradnice odkladacej plochy. Proces sa opakovane realizuje až do okamihu, kedy už nebude na odkladaciu plochu možné umiestniť ďalší objekt alebo keď bude splnená určitá podmienka ukončenia. Celý postup premiestňovania objektov prebieha automatizovane bez nutnej intervencie používateľom. Tento prístup je veľmi výhodný pre aplikácie, kde musí byť zaručená presnosť a opakovateľnosť operácií. Po skončení presúvania riadiaci systém vyhodnotí vhodnosť podmienok skladania. V prípade, že sú na odkladacej ploche prítomné všetky potrebné typy objektov, program umožní kompozíciu trojkomponentovej súčiastky robotickým ramenom. V opačnom prípade systém vyhlási chybu v podobe hlášky, ktorá sa zobrazí používateľovi v grafickom rozhraní.

Nakoľko je riadenie decentralizované, určitá časť spracovania dát a výpočtov sa vykonáva priamo v riadiacej skrinke. Ako bolo spomenuté v predošlej podkapitole, lokálnymi prvkami riadenia v skrinke sú jednodoskový počítač Raspberry Pi a vlastná doska s integrovaným mikroprocesorom STM32. Prepojenie medzi notebookom a Raspberry Pi sa uskutočňuje mobilným prístupovým bodom, čím vzniká lokálna sieť a umožňuje tak komunikovať s *Multicast.py* aplikáciou. Na počítači Raspberry Pi je spustená aplikácia *RaspberryServer.py*, ktorá zastáva dvojakú funkcionality. Prvou z nich je neustále čakanie na požiadavky od klienta. V prípade prenosu dát v podobe vypočítaných uhlov z inverznej kinematiky spustenej na notebooku sa aktivuje druhá funkcionality programu – odosielanie dát po sériovej linke do mikroprocesora STM32 s nastaveným modulačnej (prenosovej) rýchlosti na 250 000 bajtov za sekundu. Sériová komunikácia prebieha medzi týmito zariadeniami len jednosmerne. Pre reprezentáciu odosielaných dát bol špeciálne navrhnutý dátový blok, ktorý sa po prijatí do dosky STM32 evaluuje. Štruktúra dátového bloku pozostáva zo siedmich hodnôt vyjadrených reálnymi číslami. Prvou hodnotou je hlavičkový príznak s dvoma možnými stavmi – 0.0 alebo 1.0. Zvyšných šesť

hodnôt v bloku je dátových. V prípade, že je prijatý príznak nastavený na hodnotu 0.0, dáta obsiahnuté v bloku interpretujú uhly pre jednotlivé kĺby:

```
# príklad datoveho bloku - uhly pre klby robotickeho ramena
data = [0.0, 1.571, 0.0, 0.785, 0.430, 1.529, 0.0]
# príklad datoveho bloku - uchop robotickeho ramena
data = [1.0, 75.0, 0.0, 0.0, 0.0, 0.0, 0.0]
```

V opačnom prípade, ak má príznak hodnotu 1.0, program berie do úvahy len druhú hodnotu z bloku dát – predstavuje mieru zovretia úchopných čelustí, ostatné hodnoty obsiahnuté v dátovom bloku zahadzuje. Prípustné hodnoty úchopu sú z rozsahu 0.0 až 75.0, čo vyjadruje vzdialenosť medzi čelustami 70mm až 20mm. Šírka úchopu pre charakter nášho riešenia plne postačuje.

Popis hlavnej slučky programu spusteného na STM32 je nasledovný. Nižšie uvedený zdrojový kód napísaný v jazykoch C/C++ opisuje riadenie robotického ramena:

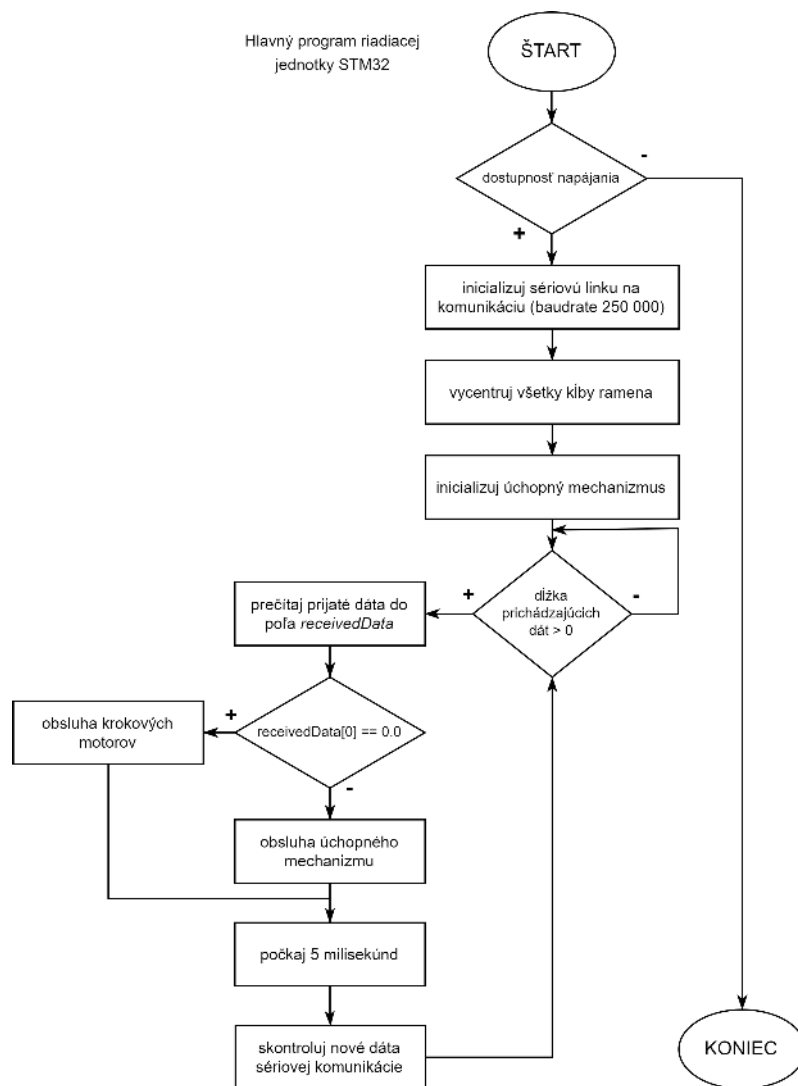
```
#include "RoboticArm.h"
RoboticArm arm = RoboticArm();

void setup() {
    arm.Initialize();
    arm.CenterJoints();
    arm.ResetGripper();
}

void loop() {
    arm.ReceiveData();
    delay(5);
}
```

Pre lepšiu štrukturalizáciu riadenia bola vyvinutá aj vlastná podporná knižnica s názvom **RoboticArm**. Vo vnútri knižnice sú uložené informácie stave kĺbov,

úchopného mechanizmu, konfigurácii sériovej komunikácie a tiež aj kompletná riadiaca funkcionálnosť. Po vytvorení globálnej inštancie triedy `RoboticArm` sa vo funkcii `setup()` najskôr inicializuje a následne prebehne centrovanie jednotlivých kĺbov spoločne s resetovaním úchopných čelustí robota na počiatočný stav. V nekonečnej slučke `loop()` napokon program riadenia robotického ramena pravidelne kontroluje príjem dát zo sériovej linky. Stabilitu hlavného cyklu zabezpečuje pozastavovanie behu programu každých 5ms . V prípade prijatia validného dátového bloku sa vykoná príslušná obslužná sada inštrukcií v závislosti od prvej hodnoty – príznaku v bloku. Ak sú prijímanými údajmi uhly pre kĺby ramena, program prepočíta rýchlosti krokových motorov tak, aby všetky započali a skončili pohyb v rovnakom čase, pričom tieto pohyby aj okamžite vykoná. Pre zjednodušenie úlohy sme neuvažovali nad zrýchleniami ani ryvom. Za predpokladu, že prijaté dáta sú určené pre uskutočnenie pohybu úchopným mechanizmom sa miera zovretia vloží do príslušnej obslužnej funkcie, ktorá následne plynule ovláda čeluste pomocou PWM signálu. Vývojový diagram na obrázku 4–13 graficky vyjadruje priebeh uvedeného obslužného kódu na STM32.



Obrázok 4–13 Vývojový diagram programu riadiacej jednotky STM32

5 Návrh a realizácia integrácie počítačového videnia do riadenia

Po skompletizovaní robotického ramena sme sa presunuli k vytvoreniu integrácie počítačového videnia do jeho riadenia. Súčasťou kapitoly je kompletný popis návrhovej časti systému počítačového videnia, ktorý zahŕňa popis stojanu na senzor Kinect s uvedením typov objektov, systém extrakcie hĺbkovej mapy a jej spracovanie, a architektúru softvérovej integrácie počítačového videnia do riadenia ramena spôsobom, ktorý umožňuje ramenu reagovať na vizuálne podnety v podobe manipulácie s objektmi.

5.1 Stojan na Kinect a popis výrobného procesu

Pri práci s počítačovým videním je potrebné mať kamerový systém vhodne umiestnený, aby bol schopný spoľahlivo pracovať. Vzhľadom k výberu senzoru Kinect pri riešení detekcie objektov v hĺbkovej mape bolo nevyhnutné zabezpečiť stojan s držiakom. Skôr než sme stojan zrealizovali boli vykonané rôzne počiatkové testy. Zisťovalo sa, v akej výške možno senzor používať na spoľahlivé snímanie a detekciu objektov. Snímací rozsah Kinect-u uvádzaný v analytickej časti práce je na tento účel postačujúci a zariadenie môže operovať v optimálnej vzdialenosti 750 mm nad zemou. Orientácia senzoru je smerovaná k zemi, pričom sa kladie veľký dôraz na dosiahnutie čo možno najlepšieho kolmého stavu. Snímaná časť podlahy pod stojanom reprezentuje pracovný priestor ramena, dostatočný na manipuláciu s objektmi. Prispôbenie obrazovej roviny ku podlahe je neskôr v realizačnej časti integrácie počítačového videnia riešené softvérovo – orezaním hĺbkového obrazu na vyhovujúce rozmery.

Konštrukcia stojanu (Obrázok 5–1) je veľmi jednoduchá, praktická a svojím tvarom pripomína šibenicu. Väčšina priemyselných riešení v praxi využíva ako stavebný materiál hliníkové profily, nakoľko sa s nimi veľmi dobre pracuje. Uplatnením po-



Obrázok 5 – 1 Stojan na senzor Kinect

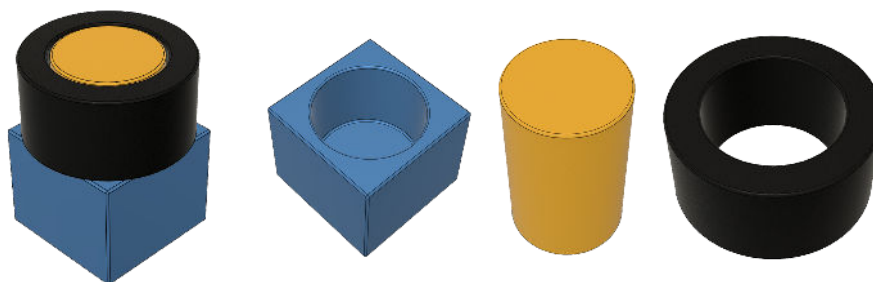
znatku sme preto celú konštrukciu stojanu navrhli práve z tohto materiálu. Stojan je zložený z piatich nezávislých segmentov od firmy Bosch o veľkosti $45\text{mm} \times 45\text{mm}$ spájaných skrutkovateľnými spojmi. Vo vrchnej časti stojanu sa nachádza hlavný trám. Do koľajníc profilu trámu sa vsúvajú špeciálne úchytové držiaky, medzi ktoré sa vkladá senzor Kinect. Najskôr boli navrhnuté v softvéri Fusion 360 rovnako ako aj robotické rameno a napokon sa realizovali opäť pomocou 3D tlače. Súčiastky sú zrkadlovo symetrické, čo zjednodušilo a skrátilo čas na návrh držiaku na Kinect. Vonkajšie rozmery stojanu sú $700\text{mm} \times 350\text{mm} \times 790\text{mm}$. Vzniknutý priestor pod senzorom Kinect má rozmery $610\text{mm} \times 350\text{mm} \times 750\text{mm}$ a poskytuje dostatok miesta na vykonávanie úkonov robotického ramena. Priestor je rozdelený na dva segmenty. Prvá časť plochy slúži robotovi na snímanie najvyššej súčiastky v kope. Druhá časť predstavuje pre robota temporárny úložný priestor pre odkladanie zosnímaných objektov kamerovým systémom. Na tomto mieste bude rameno výslednú trojdielnu súčiastku skladať.

Na pracovnej ploche pod stojanom sú v kope náhodne umiestňované 3D vytlačené telesá. Jedná sa o tri pevne stanovené vopred definované typy:

- **box** – kváder so štvorcovým pôdorysom s valcovým otvorom vo vrchnej časti,

- **cylinder** – valec s kruhovým pôdorysom,
- **toroid** – prstenec s pôdorysom medzikružia.

Súčiastky sú vytlačené v rôznych farebných prevedeniach, nakoľko pri snímaní hĺbkovou kamerou nezáleží na farbe telesa. Spojením uvedených troch telies vzniká jednoduchá trojkomponentová súčiastka symbolizujúca pomyselnú kovovú základňu (box) s oskou (cylinder) a ložiskom (toroid). Na obrázku 5–2 sa na ľavej strane nachádza ukážka zhotovenej súčiastky, napravo od nej sú znázornené jej jednotlivé časti.



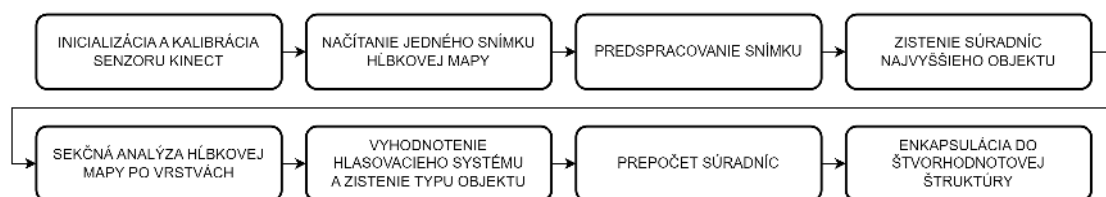
Obrázok 5–2 Zložená trojkomponentová súčiastka a jej komponenty

5.2 Návrh architektúry integrácie počítačového videnia

Aby bol zabezpečený vizuálny vnem pre integráciu počítačového videnia do riadenia je na stolný počítač napojený pomocou USB kábla senzor Kinect. Podrobný spôsob prepojenia a konfigurácia je bližšie špecifikovaná v dokumentácii záverečnej práce. Úlohou aplikácie *ComputerVisionServer.py* spúšťanej na počítači je v prípade prijatia požiadavky na rozpoznávanie zo strany klienta zosnímať hĺbkovú mapu, vykonať predspracovanie obrazu, následne detekovať a identifikovať najvyšší objekt. Výstupom zo systému je napokon štvorhodnotová dátová štruktúra, v ktorej sú obsiahnuté tri kartézské pozičné súradnice objektu x, y, z a typ objektu definovaný ako reťazec znakov.

Algoritmus počítačového videnia je možné zahrnúť do niekoľkých blokov. Celý proces je zhrnutý v jednoduchej blokovej schéme (Obrázok 5–3). Prvým krokom

je extrahovanie hĺbkovej mapy z Kinect-u v podobe dvojrozmernej matice hodnôt vzdialeností. Extrakcia prebieha pomocou knižnice *pykinect2* zapísanej v jazyku Python, ktorá uľahčuje prácu so sensorom. Výstupom jednej z funkcií knižnice je priamo požadovaná maticovo uložená hĺbková mapa. Nakoľko je Kinect umiestnený v istej výške na stojane, pokrýva väčšiu plochu než potrebujeme. Preto je matica zre-



Obrázok 5 – 3 Bloková schéma algoritmu rozpoznávania

dukovaná a prebytočné údaje sú orezané na rozmery pracovného priestoru ramena, čo zároveň urýchľuje nasledujúce výpočty. Po takto prvotne upravených údajoch ďalej nasleduje časť predspracovania obrazu. Keďže získaná hĺbková mapa vykazuje určitý stupeň zašumenia a nedokonalostí, je nutné ju predspracovať metódami počítačového videnia. Všetky nasledovné uvádzané funkcie sú použité z populárnej knižnice OpenCV, kde sú implementované. Potlačenie šumu prebieha aplikovaním Gaussovského rozostrenia konvolúciou s kernelom o veľkosti 3×3 . Fragменты a nedokonalosti na obrazovej rovine do istej miery odstránilo použitie morfolologickej operácie otvorenia. Veľmi dôležitým krokom na určenie výšky rozmiestnených objektov bola aj kalibrácia výškovej hladiny snímanej plochy. Vzhľadom k tomu, že senzor Kinect pracuje na princípe merania vzdialenosti medzi laserovým projektorom a odrazovou plochou funguje prepočet hĺbkovej hladiny inverzne. Ak je objekt umiestnený do pracovného priestoru, lúč sa odrazí skôr, čo sa prejaví ako menšia nameraná vzdialenosť. Preto je nevyhnutné počiatočné zistenie rovnice roviny pracovného priestoru. Hľadaná rovnica (5.1) v tvare

$$ax + by + cz + d = 0 \quad z = -\frac{c}{d} \quad (5.1)$$

je vypočítateľná metódou RANSAC, z ktorej pre naše účely extrahujeme len parametre c a d . Na základe nich tak určíme výšku, resp. hĺbku z . Zistenie koordinátov

najvyššieho objektu tak prebieha zistením súradníc v šedotónovej reprezentácii hĺbkovej mapy – x_{px} a y_{px} . Predstavuje miesto s globálnym minimom, tj. hodnotou najmenšej vzdialenosti od snímača. Rozdielom vzdialeností medzi nájdeným minimom a podlahovou rovinou dostávame výslednú súradnicu z_{mm} . Nájdené súradnice x_{px} a y_{px} na obrazovej rovine sú vyjadrené v pixeloch, pričom hĺbka z_{mm} je vyjadrená v milimetroch. Preto je nevyhnutné vykonať dodatočné preškáľovanie prvých dvoch hodnôt x_{px} a y_{px} z pixelov na milimetre. Zvolili sme si jednoduchú lineárnu interpoláciu zapísanú maticovo ako súčin matice a vektora, čím dostávame súradnice v milimetroch s počiatkom koordinačného systému od ľavého horného rohu. Na definitívne určenie bodu v priestore ramena je potrebný dodatočný súčet preškáľovaných súradníc so zvislým translačným vektorom. Hodnoty vo vektore vychádzajú z geometrickej podstaty pozície stojanu voči základni robota.

$$P = \begin{bmatrix} x_s & 0 & 0 \\ 0 & y_s & 0 \\ 0 & 0 & z_s \end{bmatrix} \begin{bmatrix} x_{px} \\ y_{px} \\ z_{mm} \end{bmatrix} + \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (5.2)$$

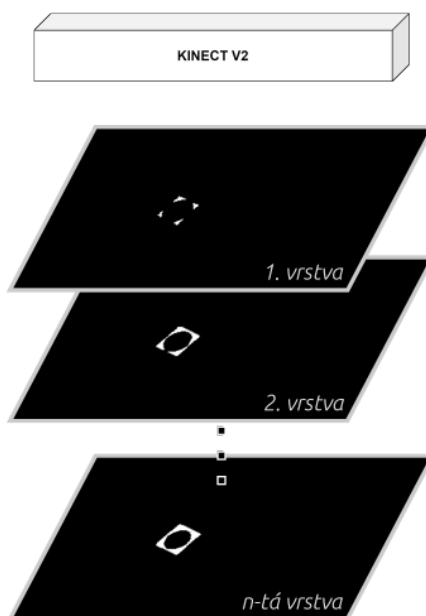
Aplikovaním prepočtového vzorca medzi hĺbkovou kamerou a pracovným priestorom získame konečnú pozíciu bodu v reálnom priestore so súradnicami $P = [x, y, z]$. Hodnoty škály pre jednotlivé osi spoločne s translačnými hodnotami vzťahu medzi relatívnymi koordinačnými rámcami Kinect-u a robotického ramena použité vo vzorci sú prehľadovo sumarizované v tabuľke 5–1.

Parameter	Hodnota	Parameter	Hodnota
x_s	1.950	x_t	150.0
y_s	1.925	y_t	–305.0
z_s	1.000	z_t	–40.0

Tabuľka 5–1 Hodnoty použité pri prepočte súradníc

Poslednou fázou celého algoritmu je identifikácia typu najvyššieho detegovaného objektu. Etalón detegovateľných typov tvorí trojica *box*, *cylinder* a *toroid*. K riešeniu

rozpoznávania nás inšpiroval hlasovací systém Houghovej transformácie. Zjednodušili sme jeho myšlienku a adaptovali ju potrebám nášho programu. Hlasovanie prebieha postupným prehľadávaním a analýzou vrstiev hĺbkovej mapy (Obrázok 5–4). Od najvyššie nájdeného bodu postupujeme nižšie o n vrstiev tak, ako je znázornené na obrázku. V každej z nich sa vykoná určitá séria operácií, ktorá prispieva do hlasovacieho systému. Operácie zahŕňajú vyhodnotenie počtu blobov, počet nájdených úsečiek spoločne s ich dĺžkami a počet nájdených kružníc s ich polomerami. Hlasuje sa



Obrázok 5–4 Schéma prehľadávania hĺbkovej mapy po vrstvách

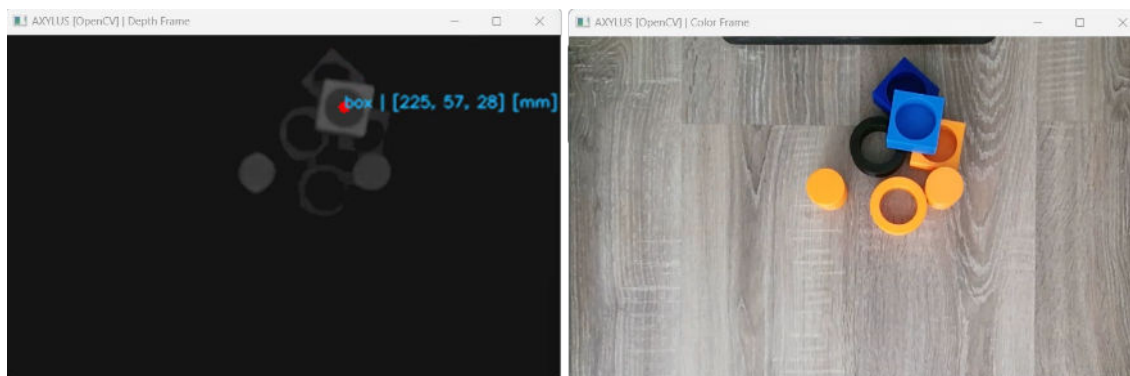
pomocou postupného pripočítavania heuristicky zvolených desatinných čísel z rozsahu $\langle -5.0; 5.0 \rangle$ každému z trojice objektov na základe určitých programovo vopred zadaných podmienok. O výsledku rozpoznávania rozhoduje objekt s najvyššie dosiahnutou hodnotou hlasovacieho čísla.

5.3 Realizácia integrácie počítačového videnia do riadenia

Robotické rameno a jeho riadenie je plne nezávislé od predstaveného algoritmu počítačového videnia. Grafické užívateľské rozhranie je schopné ovládať virtuálny,

ale aj fyzický model robota. Integráciou algoritmu sme rozšírili riadenie o reakciu na vizuálne vnemy prijímané prostredníctvom senzora Kinect a tým aj grafické rozhranie o ďalší prvok v menu (viac informácií v kapitole 6). Riadiaca infraštruktúra už nereaguje len na zmeny vykonávané užívateľom, ale taktiež má možnosť komunikovať priamo s aplikáciou *ComputerVisionServer.py*, ktorá má na starosti rozpoznávanie objektov.

Po vložení senzoru Kinect do stojanu, počiatočnej kalibrácii roviny pracovného priestoru a spustení aplikácie sme boli schopní rozpoznávanie otestovať. Na pracovnú plochu pred ramenom sa náhodne rozmiestnila sada objektov s rôznou farebnosťou. Objekty boli zámerne vytlačené rozličnými farbami, aby sa demonštrovala schopnosť invariančného rozpoznávania, nakoľko Kinect pracuje na vlnovej dĺžke inej ako je viditeľné svetlo. Pre lepšiu predstavu, ako Kinect vníma vizuálne podnety, boli v rámci kódu naprogramované aj dve funkcie na zobrazovanie hĺbkovej mapy interpretovanej ako šedotónový obraz a zobrazovanie výstupu z farebnej kamery. Túto možnosť nám poskytla funkcia z knižnice OpenCV. Po spustení testovacieho skriptu sa na počítači otvorili dve nové okná (Obrázok 5–5). Už na prvý pohľad je zrejmé, že farebný obraz z kamery podlieha istému radiálnemu skresleniu (tzv. rybie oko). Naproti tomu je voči tomuto typu skreslenia hĺbkový senzor vďaka svojim konštrukčným vlastnostiam robustný, čo je pri riešení rozpoznávania značnou výhodou. Do obrazu s vykresľovaním šedotónovej hĺbkovej mapy sme vpisovali aj typ nájdeného objektu



Obrázok 5–5 Hĺbková mapa (vľavo) a farebný obraz (vpravo) zo senzoru Kinect

spoločne s tromi súradnicami udávané v milimetroch. Interpretácia hĺbkovej mapy ako šedotónový snímok spôsobil, že povrchy bližšie pri senzore sa javia svetlejšie ako povrchy, ktoré sa nachádzajú ďalej. Z uvedeného dôvodu je na obrázku podlaha zobrazená čiernou farbou. Červený bod symbolizuje nájdený centroid blobu v oblasti s najväčšou vzdialenosťou od roviny pracovného priestoru. Respektíve sa jedná o oblasť, ktorá je k hĺbkovému snímaču najbližšie.

Na konci algoritmu sa objekt zabalí do inštancie triedy *RecognizedObject*, ktorá obsahuje jeho typ a súradnice. Z hľadiska optimalizácie a zvýšenia pravdepodobnosti správneho rozpoznania je algoritmus vykonaný určitý počet krát za sebou, pričom výstupy sú uložené do temporárneho pomocného poľa. Do úvahy sa napokon berie výsledok s najfrekvencovanejším výskytom daného typu objektu v tomto poli. Dôvodom je nestabilita fyzikálnych podmienok akou je napríklad osvetlenie v miestnosti, odrazy lúčov od lesklých povrchov či iné externé faktory ovplyvňujúce štruktúru hĺbkovej mapy. Pri vytvorení požiadavky z grafického užívateľského rozhrania je mu ako odpoveď spätne odosielaná inštancia *RecognizedObject* interpretovaná ako serializované asociatívne pole.

Akonáhle bol algoritmus dostatočne otestovaný a pripravený na použitie, mohli sme prejsť k integrácii počítačového videnia do riadenia robotického ramena. Vzhľadom k tomu, že úlohou diplomovej práce je poskladanie trojkomponentovej súčiastky s využitím robotického ramena sa určitým spôsobom prepojil algoritmus so zvyšnou časťou riadenia. Pracovná plocha pred manipulátorom bola rozdelená do dvoch oblastí:

1. **oblasť skladania** výslednej súčiastky,
2. **oblasť s rozmiestnenými objektmi** náhodne umiestňovanými po celej ploche snímanej Kinectom s výnimkou oblasti skladania.

Vyhradené miesto, kde si robotické rameno postupne temporárne ukladá presunuté objekty je hĺbkovou kamerou ignorované, aby nedošlo ku kolízii s hlavnou časťou rozpoznávacieho algoritmu. Je rozdelené do štyroch segmentov, označených písmenami

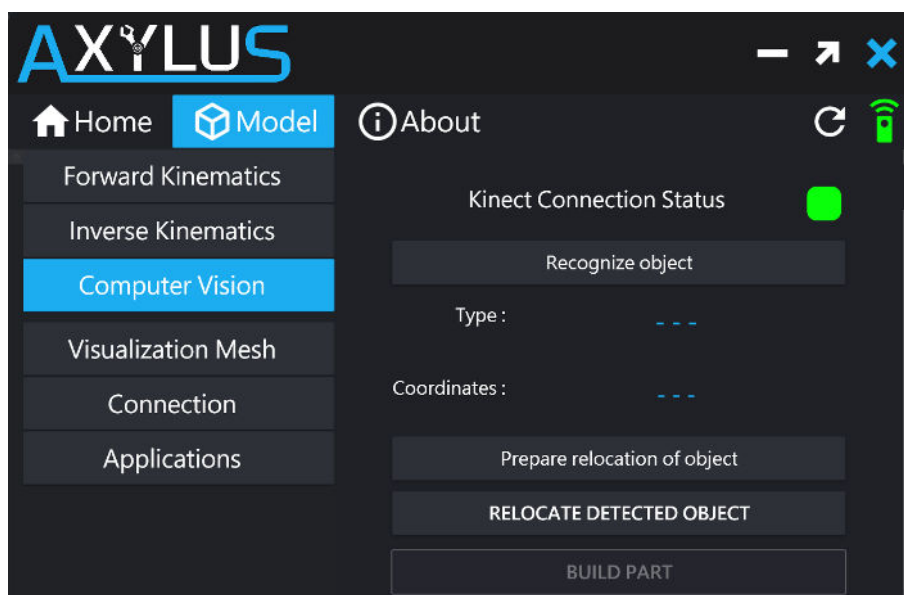
A až D, a nachádza sa v ľavej hornej časti obrazu. Na obrázku 5–6 je odkladacia plocha graficky zobrazená červeným orámovaním spolu s označením konkrétnych segmentov. Jedna pozícia je redundantná a poskytuje opravný pokus v prípade, že riadiaci program po troch presunoch nezistí prítomnosť všetkých typov objektov na ohraničenej ploche potrebných na poskladanie súčiastky. V grafickom



Obrázok 5–6 Grafické znázornenie úložného priestoru

užívateľskom rozhraní je v záložke počítačového videnia umiestnená sada tlačidiel (Obrázok 5–7). Každé tlačidlo zastupuje určitú funkcionality. Obsluha riadiaceho algoritmu presúvania a skladania je riešená semiautomatcky a sekvenčne. V poradí prvým tlačidlom *Recognize object* sa najskôr rozpozná najvyšší objekt v ploche a informácie o ňom sa zobrazia v textovej podobe pod ním. Ďalšie tlačidlo *Prepare relocation of object* slúži na výpočet inverznej kinematiky k bodu nájdeného v predošlom kroku. Zároveň sa aktualizuje aj vizualizácia, aby sa vizuálne verifikovala správnosť výpočtu a nedošlo tak k poškodeniu fyzického modelu robota. Predposledným tlačidlom dostupným v menu užívateľského rozhrania je *RELOCATE DETECTED OBJECT*, ktoré po stlačení vygeneruje sekvenciu inštrukcií. Rameno na základe nich presunie svoj koncový bod s úchopným mechanizmom ponad objekt, následne ho chytí medzi čeluste a presunie do oblasti skladania do jedného zo segmentov. Objekt si program uloží do pamäte a segment označí ako obsadený. Uvedené kroky uživa-

teľ opakuje dovtedy, kým nie sú odkladacej ploche obsadené všetky bunky alebo pokiaľ nie je splnená podmienka. Za predpokladu splnenia podmienok skladania sa aktivuje aj posledné tlačidlo *BUILD PART* a umožní vykonať skladáciu procedúru. Program v pamäti nájde súradnice objektu **box**, ktorý je podstavou súčiastky. Nasleduje úchopno-presúvacia rutina objektu **cylinder** so vsunutím do vnútra tela podstavy. Finálnym krokom je vloženie objektu **toroid** do objektu **cylinder**, čím sa ukončí proces skladania a robotické rameno sa vráti do počiatočného stavu. Ak na-



Obrázok 5–7 Okno grafického užívateľského rozhrania pre modul počítačového videnia

stane situácia, pri ktorej program vyhodnotí aj napriek plnej obsadenosti úložného miesta neschopnosť súčiastku poskladať, užívateľské rozhranie zahlásí chybu v podobe hlášky, čím upozorní o tomto stave používateľa.

6 Vizualizácia a grafické užívateľské rozhranie

Súčasťou riešenia problematiky riadenia navrhovaného robotického ramena bolo vytvorenie vizualizácie a grafického užívateľského rozhrania. Vizualizačný softvér umožňuje vykonať testy kinematických výpočtov a overiť správnosť priamej a inverznej kinematiky ramena. Zároveň poskytuje ochranu konštrukcie fyzického modelu pred poškodením, pretože sú všetky pohyby robota najprv testované virtuálne. Naproti tomu je užívateľské rozhranie centrálnym ovládačom pre robotické rameno s možnosťou interaktívne pracovať s virtuálnym, ale aj fyzickým modelom manipulátora. Rozhranie má navyše integrované ovládanie rozpoznávacieho algoritmu spôsobom, ktorý bol popísaný na konci predošlej kapitoly.

6.1 Vizualizácia robotického ramena

Prerekvizitou vizualizácie nami skonštruovaného robotického ramena je popis kinematickej štruktúry. Jej jednotlivé zložky boli bližšie spísané v podkapitole 4.1.2. Hodnoty rotácií a translácií je však potrebné počítačovo interpretovať vhodným súborovým formátom. Pre tieto účely je najvýhodnejší značkovací jazyk eXtended Markup Language (XML), ktorý štrukturalizuje kinematickú štruktúru. Aby sme oddelili vizualizačné prvky ramena od jeho kinematického zretazenia, vznikla potreba vytvorenia dvoch separátnych súborov.

Prvý súbor s názvom **KinematicModel.xml** sa skladá zo značky *Chain* v rámci ktorej sú definované translácie a rotácie, čím popisuje celú štruktúru manipulátora. V súbore sa taktiež nachádzajú aj dodatočné parametre pre nastavenie inverznej kinematiky, akými sú počet iterácií algoritmu a tlmiaca konštanta. V úryvku súboru sa nachádza príklad definície translačnej časti modelu značkou *Link* a rotačnej časti značkou *Joint*. Translácia je zložená z troch translačných zložiek x, y, z vyjadrených dĺžkou v milimetroch. Rotácia je naopak popísaná jednou z osí x, y alebo z a hodnotou uhla v radiánoch v počiatočnom kinematickom stave. Celý súbor je čítaný skriptom v jazyku Python, ktorý po otvorení súboru spracuje obsiahnuté údaje

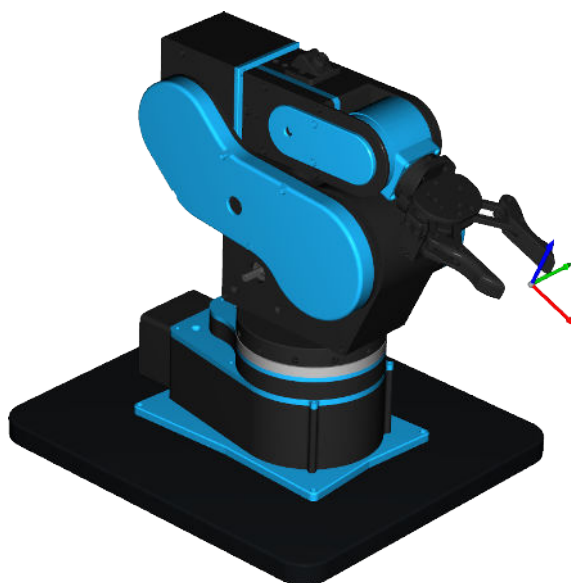
do takej dátovej štruktúry, aby s ňou bolo možné vykonávať kinematické výpočty:

```
<KinematicModel name="Axylus">
  <InverseKinematics iterations="10" damping="0.0174533"
    deviation="1.0"/>
  <Chain name="Chains">
    <Link x="0.0" y="0.0" z="99.50">
      <Joint axis="z" angle="0.0"/>
    </Link>
    ...
  </Chain>
</KinematicModel>
```

Druhý súbor pomenovaný ako **VirtualModel.xml** je špecificky zameraný na popis priradení STL súborov komponentov ku spomínanej kinematickej štruktúre:

```
<VirtualModel name="Axylus">
  <MeshDirectory path="Files/STL/Model"/>
  <RigidGroup name="Base">
    <RigidBody>
      <Translations x="0.0" y="0.0" z="0.0"/>
      <Rotations x="0.0" y="0.0" z="-1.5707"/>
      <Mesh title="Base Bottom Bottom" visible="True">
        <Translations x="0.0" y="0.0" z="0.0"/>
        <Rotations x="0.0" y="0.0" z="0.0"/>
        <TriangleMesh fileName="Base Bottom Bottom.stl"/>
        <Color r="0.11" g="0.68" b="0.94"/>
      </Mesh>
      ...
    <RigidBody>
  </RigidGroup>
```

Týmto spôsobom sa zaručí prepojenie počítačom navrhnutých 3D objektov s kinematikou. Súbor zahŕňa niekoľko XML elementov. Značka *MeshDirectory* určuje cestu k priečinku so súbormi modelu manipulátora. Na popis rigidnej skupiny telies slúži značka *RigidGroup*. Jedna skupina môže obsahovať viacero telies. Každé teleso *RigidBody* je definované transláciami a rotáciami v priestore a obsahuje značku *Mesh*. Mriežka je špeciálne navrhnutý typ elementu s vlastným názvom a nastaviteľnou viditeľnosťou. V rámci nej je obsiahnutý zdrojový stereolitografický súbor ku konkrétnemu telesu, nastavenie farby telesa a rovnako ako aj celé teleso je mriežke možné prideliť internú transláciu a rotáciu. Úryvok z XML súboru je ilustračne uvedený nižšie. Súbor je opäť ako v prípade kinematického modelu čítaný Python skriptom, avšak s rozdielom, že dáta sú po načítaní vložené do funkcií vizualizačnej knižnice Open3D [25]. Táto rozsiahla knižnica poskytuje veľa rôznych nástrojov vrátane užitočných funkcií slúžiacich na vizualizáciu. Vstupom je zoznam geometrických objektov, ktoré vykresľuje vo vytvorenom okne pomocou rozhrania OpenGL. Importovaných bolo spolu osemnásť STL súborov, z ktorých sa následne zostavil virtuálny model robotického ramena. Pod ramenom je umiestnený podľa skutočnej predlohy aj virtuálny podstavec. Pracovný priestor pod stojanom na senzor Kinect symboli-



Obrázok 6 – 1 Vizualizácia ramena realizovaná knižnicou Open3D

zuje tenká biela platňa. Agregáciou vymenovaných častí do jedného celku dostávame výslednú podobu vizualizácie (Obrázok 6–1). Podľa pojmov z oblasti Industry 4.0 by sme mohli túto vizualizáciu pomenovať aj ako digitálne dvojča. Kinematický model spustený na počítači sa do istej miery priamo odzrkadľuje na uhloch fyzického robotického ramena. Riadiaca skrinka robota komunikuje s kinematickým modelom v reálnom čase a prijíma dátové bloky spomenuté v kapitole 4, na základe ktorých vykonáva akčné zásahy. Veľkou výhodou virtuálnej reprezentácie modelu ramena je skutočnosť, že nemôže dôjsť k fyzickému poškodeniu súčiastok. Spoločne s grafickým rozhraním poskytuje možnosť pracovať s robotom bez prítomnosti jeho fyzickej podoby, čo okrem iného spôsobuje aj urýchlenie testovania kinematických výpočtov – priamej a inverznej kinematiky (kapitola 2).

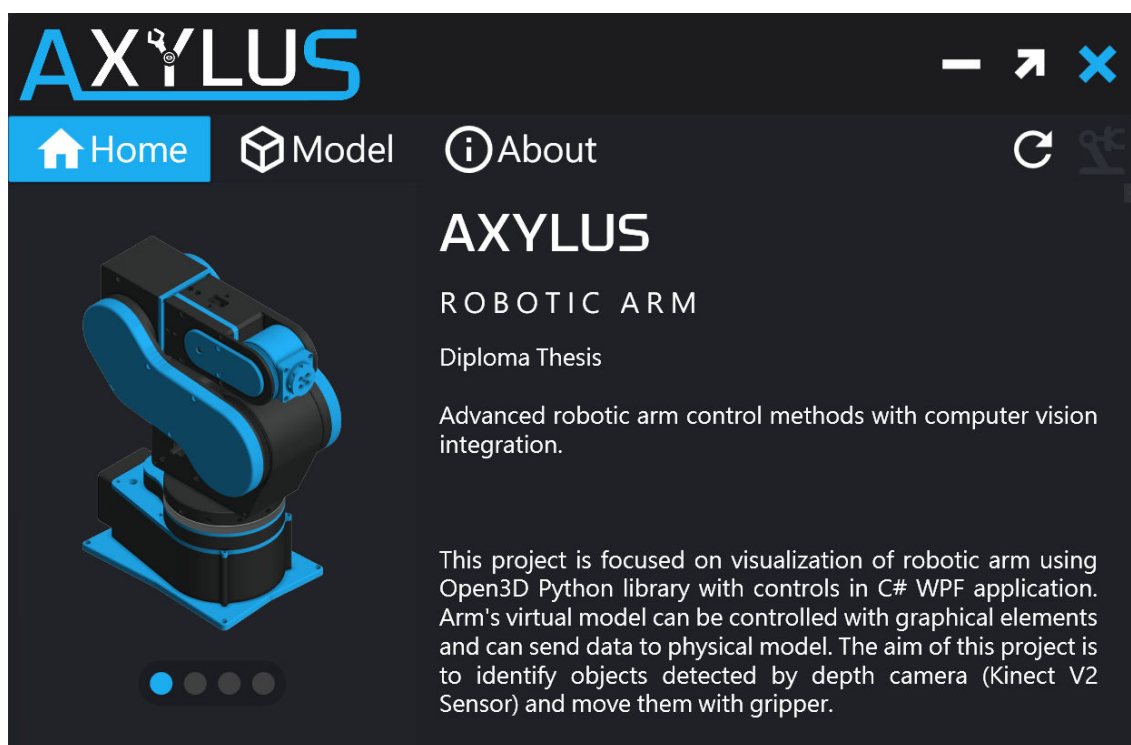
Celú vizualizáciu zabezpečuje aplikácia *VisualizationServer.py*, ktorá je spúšťaná priamo z grafického rozhrania. Kliknutím na tlačidlo sa vykoná príkaz spôsobujúci otvorenie príkazového riadku a pomocou neho sa aplikácia spustí, čím vytvorí novú inštanciu okna s vizualizáciou v Open3D. Aplikácia následne čaká na jednotlivé užívateľské akcie a reaguje na ne aktualizáciou stavu virtuálneho modelu. Podrobnejší popis interakcie medzi vizualizáciou a grafickým rozhraním je spísaný v nasledujúcej podkapitole.

6.2 Grafické užívateľské rozhranie

Pretože riadenie robotického ramena v bakalárskej práci bolo užívateľsky neprívetivé, rozhodli sme sa ho unifikovať do jednotného rozhrania a vyvinúť samostatnú desktopovú aplikáciu spustiteľnú pod operačným systémom Windows. Spoločnosť Microsoft v roku 2006 vydala rámec užívateľského rozhrania WPF – Windows Presentation Foundation [26] programovateľný v jazyku C#. Jeho fundamentálnymi zložkami sú elementy (značky) založené na XML jazyku štrukturalizované do súboru, ktorý sa nazýva Extensible Application Markup Language (XAML). Táto technológia umožňuje podľa potreby vytvárať rôzne grafické užívateľské rozhrania

(ďalej GUI) vďaka využitiu elementárnych grafických prvkov, akými sú napríklad tlačidlá, textové polia, posuvníky, rozbalovacie ponuky, dátové mriežky a rôzne ďalšie. Výhodou je pomerne rýchly a pohodlný vývoj ľubovoľného rozhrania podľa určitých definovaných požiadaviek. Z uvedených dôvodov sme sa rozhodli pre výber tohto nástroja schopného pokryť všetky nami zvolené kritériá. Zahŕňa to vytvorenie manuálneho ovládania priamej kinematiky, inverznej kinematiky a úchopného mechanizmu virtuálneho i fyzického modelu ramena, zobrazovanie a skrývanie jednotlivých komponentov konštrukcie reprezentatívnou tabuľkou, ovládanie modulu počítačového videnia s rozpoznávaním objektov v hĺbkovej mape a tiež aj priame spúšťanie serverových aplikácií skrz toto rozhranie.

Samotné GUI je rozdelené do troch hlavných záložiek umiestnených v hornom paneli (Obrázok 6–2). V záložke *Home* sa nachádza stručný popis diplomovej práce a grafický náhľad robotického ramena. Záložka *Model* je najpodstatnejšou časťou



Obrázok 6–2 Grafické užívateľské rozhranie pre robotické rameno

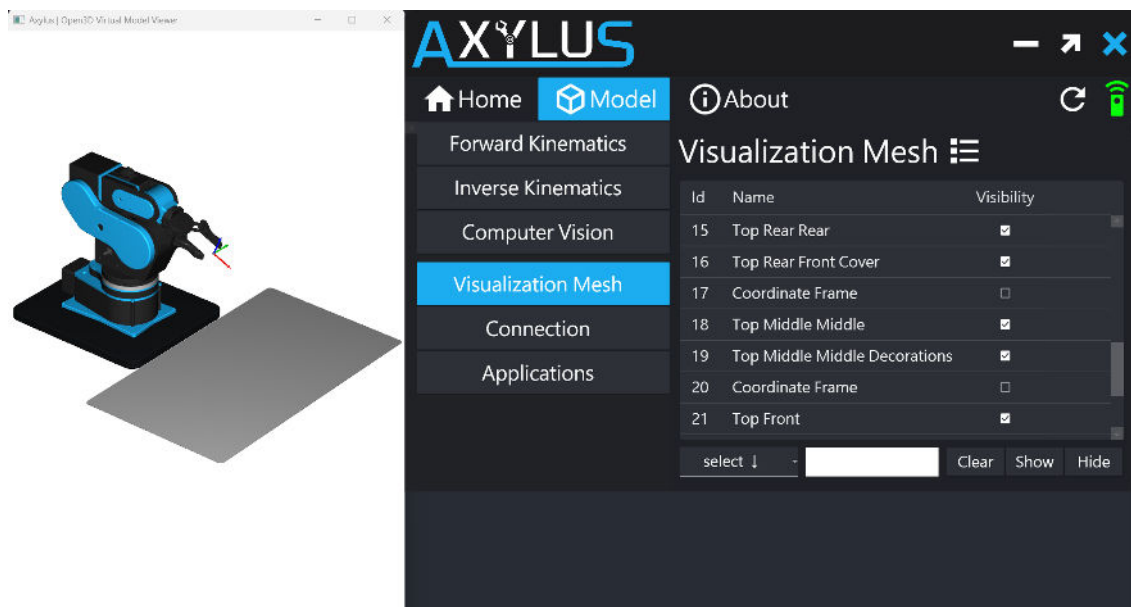
GUI, nakoľko práve v nej sú obsiahnuté všetky vyššie spísané ovládacie prvky robota.

Pozostáva z niekoľkých položiek:

- **Forward Kinematics** – okno je zostavené z viacerých grafických prvkov, pričom slúži na ovládanie jednotlivých kĺbov robotického ramena,
- **Inverse Kinematics** – v okne sa nachádza šesť políčok, tri slúžia na popis pozície bodu v priestore v milimetroch, ďalšie tri slúžia na opis jeho orientácie v radiánoch,
- **Computer Vision** – okno zahŕňa niekoľko tlačidiel, bližšie spísaných v predošlej kapitole (kapitola 5),
- **Visualization Mesh** – v rámci okna je umiestnená tabuľka so zoznamom STL súborov používaných vo vizualizácii, obsahuje možnosť skrývať alebo odkrývať jednotlivé časti virtuálneho modelu ramena,
- **Connection** – okno s textovým polom pre príslušnú IP adresu a port, a tlačidlom pre pripojenie,
- **Applications** – vnútri okna sú rozmiestnené tri tlačidlá - každé z nich spúšťa jeden zo serverov.

Postup spúšťania je pomerne jednoduchý a skladá sa z troch operácií. Ako prvé sa musia pozapínať jednotlivé aplikačné servery v menu *Applications*. Programovo je na pozadí pre každú aplikáciu otvorený príkazový riadok (CLI), v ktorom je spustený Python skript. Na jednodoskovom počítači Raspberry Pi je nutné tento krok vykonať manuálnym spustením aplikácie *RaspberryServer.py*. Na poradí spúšťania aplikácií *VisualizationServer.py* a *ComputerVisionServer.py* a *RaspberryServer.py* nezáleží, avšak *Multicast.py* musí byť z dôvodu komunikačnej architektúry spúšťaný ako posledný. V prípade, že sú aplikácie pripravené, je možné pristúpiť k pripojeniu aplikácie *Multicast.py* ku GUI. Prostredníctvom nej sa hromadne navzájom prepoja všetky ostatné servery, čo umožní GUI s nimi komunikovať. Pripojenie prebieha zadáním príslušnej IP adresy a portu v GUI sekcii *Connection*. O úspešnom pripojení

informuje používateľa zelený indikátor v pravom hornom rohu okna. Takto pripravené rozhranie (Obrázok 6–3) umožňuje vykonávať rozličné úkony. Bližšie špecifikácie, popis programovej časti rozhrania, snímky jednotlivých okien sa nachádzajú v dokumentácii záverečnej práce.



Obrázok 6–3 Grafické užívateľské rozhranie pripravené na použitie

7 Vyhodnotenie implementovaného riešenia

Finálnym štádiom celého projektu bolo overenie funkčnosti vypracovaných častí. Splnenie zadanej úlohy si vyžadovalo sériu krokov, ktoré sú navzájom hierarchicky previazané. Skladanie súčiastky je podmienené správnym rozložením objektov na odkladacej ploche. Pozície týchto objektov na ploche sú podmienené správnym úchopom a presunom detegovaného objektu. Aby sme zabezpečili inkrementálnu formu testovania, bolo vyhodnotenie riešenia rozčlenené do troch fáz. Spočiatku sa verifikovala správnosť kinematických výpočtov a schopnosť ramena

7.1 Vyhodnotenie robotického ramena

Robotické rameno je možné testovať vo viacerých aspektoch. Môžu sa vykonať testy jednotlivých častí ramena – overuje sa správna funkčnosť každého komponentu ramena izolovane. Pri vykonávaní testov sa kontroluje a sleduje, ako jednotlivé časti reagujú na pokyny z riadiaceho systému. Komplexnejším testovaním je zisťovanie, s akou presnosťou rameno operuje. Okrem validácie funkčných častí robota je veľmi podstatné skúmať fyzické zaťaženie. V prípade, že sa robotické rameno používa na manipuláciu s ťažkými bremenami. Je potrebné skontrolovať, ako dobre zvláda fyzickú záťaž a aká je nosnosť v kontexte blízkych či vzdialených pozícií od základne ramena.

Z uvedených testovaní sme si zvolili meranie presnosti koncového bodu ramena. Zisťuje sa tým schopnosť inverznej kinematiky nadobudnúť zadaný bod. Experimentálne overovanie prebiehalo náhodným vygenerovaním množiny bodov v rámci určitého vymedzeného priestoru pred ramenom. Generované hodnoty boli pre os $x \in \langle 180.0; 400.0; \rangle$, pre os $y \in \langle -270.0; 270.0; \rangle$ a pre os $z \in \langle 0.0; 500.0; \rangle$. Vykonali sme 8 nezávislých experimentov, pričom sme si ako metriku zvolili euklidovskú vzdialenosť medzi zadaným očakávaným bodom a skutočným nadobudnutým bodom koncového efektora manipulátora. Dosiahnuté výsledky meraní sú uvedené v tabuľke 7–1. Z uvedených meraní sa zistilo, že priemerná odchýlka medzi očakávaným a sku-

Očakávaný bod [mm]	Skutočný bod [mm]	Vzdialenosť [mm]
[194, 126, 247]	[190, 110, 251]	17
[197, -27, 52]	[210, -34, 55]	15
[371, -59, 71]	[385, -50, 100]	19
[212, 116, 283]	[220, 115, 280]	9
[395, -8, 103]	[390, -10, 105]	6
[291, 40, 256]	[310, 35, 265]	22
[229, -59, 110]	[250, -63, 120]	24
[213, 43, 260]	[235, 50, 265]	24

Tabuľka 7 – 1 Experimentálne testovanie presnosti koncového bodu

točným bodom sa pohybuje na úrovni 17mm. Prejavené nepresnosti spôsobila najmä fyzická vôľa 3D vytlačených komponentov, pretože sú vyhotovené z pomerne mäkkého plastu, čo zapríčiňovalo ohyb funkčných dielov. Robot sa z počiatočnej zloženej polohy dostal do určitej kinematickej polohy na základe vypočítaných uhlov kĺbov.



Obrázok 7 – 1 Robotické rameno v určitej kinematickej polohe

Po ukončení pohybu zostali všetky motory zabrzdené, čím sa mohlo prejsť k vykonaniu merania meracím pásmom – metrom. Na obrázku 7–1 je znázornená ukážka, ako vyzerá rameno po nadobudnutí jedného zo zadaných bodov. Rotačné zložky tohto bodu boli nastavené tak, aby efektor (úchopný mechanizmus) smeroval kolmo k zemi podobne, ako pri uchopovaní a presúvaní objektov.

7.1.1 Testovanie úchopu a presunu objektu

Ďalšie testovanie robotického ramena sa zameriavalo na úchop a presun objektov, čím sa preverovala schopnosť ramena odoberať objekty z kopy. Geometrické vlastnosti každého z trojice objektov vyhovujú požiadavkám uchopovania pomocou čelustí. Rozostup medzi čelustami poskytuje dostatočný priestor aj pre najväčší objekt – toroid. Pôdorys cylindra a toroidu je kruhový, čo umožňuje mechanizmu previesť spoľahlivý úchop. Naproti tomu je box v pôdoryse štvorcový, čím podlieha afinnej transformácii, konkrétne rotácii v osi z . Preto záleží, v akom uhle je na pracovnej ploche umiestnený. Hmotnosť objektov sa pohybuje rádovo v gramoch pre ich materiálové vlastnosti (sú vyhotovené z plastu). Nepredstavujú preto pre úchopný mechanizmus ani rameno veľkú manipulačnú záťaž. V tabuľke 7–2 sú uvedené výsledky experimentálneho testovania úchopu a presunu na odkladaciu plochu. Vykonaných bolo 24 nezávislých pokusov, 8 pokusov pre každý typ objektu. Zistili sme, že box aj napriek svojmu tvaru nespôsobil úchopnému mechanizmu závažné problémy. Problematické boli jedine prípady, kedy bola orientácia boxu v uhle $45^\circ \pm 5^\circ$ voči referenčnému koordinačnému rámcu pracovnej roviny. Naopak cylinder a toroid sa pri niektorých pokusoch nepodarilo robotickému ramenu úspešne uchopiť,

–	Úspešné pokusy	Neúspešné pokusy
box	7	1
cylinder	6	2
toroid	6	2

Tabuľka 7–2 Tabuľka úspešnosti úchopu objektov

nakoľko došlo k chybnému úchopu, respektíve sa objekt z čelustí vyšmykol. Takýto typ poruchy sa prejavoval najmä v prípadoch, kedy sa koncový bod ramena nenachádzal v oblasti centroid-u pôdorysu telesa, ale mimo neho. Pre ukážku fungovania úchopu objektu obrázok 7–2 znázorňuje robotické rameno s cylindrom identifikovaným ako najvyšší v kope.



Obrázok 7–2 Ukážka úchopu objektu mechanizmom

7.2 Vyhodnotenie rozpoznávania objektov

Vzhľadom k modularite nami navrhnutého riešenia je možné algoritmus počítačového videnia testovať nezávisle od robotického ramena. Proces experimentálneho overovania rozpoznávania najvyššieho objektu nájdeného na pracovnej ploche prebiehal pomocou náhodného umiestňovania rôzneho počtu objektov do plochy. Vykonaných bolo spolu 30 pokusov – pre každý typ objektu bolo vyhradených 10 pokusov. Naľko bol každý rozpoznávací test vykonávaný s rozličným počtom objektov s variabilnou hustotou ich rozmiestnenia, skúmali sme aj možné vplyvy na náš algoritmus

rozpoznávania. Korelácia medzi počtom objektov a výsledkami sa žiadna nezistila, pretože algoritmus je nastavený len na vyhľadávanie a zisťovanie najvyššieho objektu v kope. Niektoré problémy sa avšak objavili pri objektoch v podobnej výške a tesnou blízkosťou pri sebe. Vrstvové prehľadávanie implementované v algoritme zachycovalo aj úsečky a kružnice z okolia najvyššie nájdeného bodu, čo zapríčiňovalo nesprávnu identifikáciu typu objektu. Sumarizácia dosiahnutých výsledkov je uvedená v konfúznej matici (Obrázok 7–3). Analýzou matice sa zistilo, že cylinder mal najmenej stabilný rozpoznávací výsledok. Kategorizácia boxu vykazovala najlepšie výsledky s 90% úspešnosťou. Objekt cylindra sa úspešne rozpoznal v 60% prípadov

		Predikovaná trieda		
		BOX	CYLINDER	TOROID
skutočná trieda	BOX	9	1	0
	CYLINDER	2	6	2
	TOROID	3	0	7

Obrázok 7–3 Konfúzna matica rozpoznávacieho algoritmu

v dôsledku podobného polomeru kružnice nájdeného pri boxe alebo toroide. Toroid mal vyššiu, 70% úspešnosť a algoritmus si tento objekt v 30% prípadov zamieňal za box.

7.3 Vyhodnotenie skladania trojkomponentovej súčiastky

Problematika skladania súčiastky by sa dala rozdeliť na dva procesy, ktoré pri tom prebiehajú. Najprv sa testovala schopnosť ramena úchopným mechanizmom dvíhať, prenášať a ukladať objekty do vopred určených buniek odkladacej plochy. Po vykonaných experimentoch sa mohlo prejsť k overovaniu skladania trojkomponentovej

súčiastky za predpokladu splnenia podmienky prítomnosti všetkých troch typov objektov na odkladacej ploche.

7.3.1 Testovanie úspešnosti skladania

Aby sme mohli úspešnosť skladania nejakým spôsobom vyhodnotiť, zvolili sme si kvantitatívny prístup testovania. Vytvorili sme tabuľku 7–3 s trojicou stĺpcov, ktoré predstavujú počty kompletných, čiastočných a neúspešných skladaní. Po vykonaní 20 pokusov sme výsledky postupne zapisovali do tabuľky podľa toho, ako vykonaný pokus dopadol. Za čiastočné poskladanie sa považuje taký stav, pri ktorom sú súčiastky rozmiestnené a pripravené na skladanie, no ďalej rameno nebolo schopné súčiastku poskladať tak, ako je znázornené na obrázku 7–4. Neúspešným skladaním môže byť pokus s chybnou manipuláciou s objektami, chybné rozpoznanie, prípadne skladanie nebolo možné vykonať v dôsledku nedostatočného množstva materiálu na skomponovanie súčiastky. V procese testovania mohli nastať aj kolízne prípady, kedy sú na odkladacej ploche prítomné dva boxy – súčiastkové základne.



Obrázok 7–4 Ukážka poskladanej trojkomponentovej súčiastky

Programovo je tento stav ošetrený automatickým označením v poradí prvého nájdeného výskytu boxu v úložnom priestore za základňu súčiastky. Z uvedenej tabuľky

Počet kompletných skladaní	Počet čiastočných skladaní	Počet neúspešných skladaní
12	4	4

Tabuľka 7 – 3 Tabuľka úspešnosti poskladania súčiastky

vyplýva, že robotické rameno bolo schopné úspešne poskladať súčiastku so 60% úspešnosťou, pričom zvyšných 40% všetkých vykonaných pokusov rameno súčiastku zložilo buď čiastočne alebo neúspešne. Dosiahnuté výsledky robotického systému sa dajú postupne vylepšovať pri dostatočnom množstve času na doladenie kritických častí procesu, ako je presnejší rozpoznávací algoritmus alebo výmena objektov za hmotnejšie verzie.

Záver

Cieľom tejto diplomovej práce bolo navrhnuť, vymodelovať a skonštruovať fyzický model ramena, naprogramovať jeho riadenie a rozšíriť ho o prácu s obrazovou informáciou získavanou zo senzora Kinect. Nakoľko skúsenosti s robotickými ramenami sme nadobudli už počas bakalárskej práce zameranej na túto problematiku, počítačnou úlohou práce bolo sústrediť sa na počítačové videnie a architektúru celého riadenia. Spočiatku bolo nutné zoznámiť sa so senzorom Kinect v2 a premyslieť, ako bude fungovať riadiaca infraštruktúra robota. Po počiatočných testoch a prepojení senzora s programovacím jazykom Python sme začali s návrhom konštrukcie robotického ramena a stojanu na Kinect. Fázu prototypovania neskôr nahradila fáza fyzického zhotovovania pomocou 3D tlače s následným pospájaním do funkčného celku. Vzhľadom k použitiu krokových motorov ako aktuálnych členov pre kĺby, vznikla nutnosť vytvoriť riadiacu skrinku s kompletnou elektronikou zabezpečujúcou riadenie na najnižšej úrovni. Súčasťou práce bolo taktiež navrhnuť, odkiaľ bude manipulátor riadený. Prišli sme s decentralizovanou architektúrou, kde je časť operácií vykonávaná na počítači, a istá časť vykonávaná lokálne v priamo riadiacej jednotke. Celé riadenie napokon využíva klient–server architektúru a beží pod TCP protokolom, ktorý zabezpečuje výmenu dát medzi softvérom spusteným na počítači a riadiacim programom v skrinke.

Navyše sme vytvorili aj virtuálnu reprezentáciu ramena vo forme vizualizácie a grafické používateľské rozhranie poskytujúce interaktívne ovládanie robota ez prítomnosti jeho fyzickej podoby. V rámci rozhrania používateľ dokáže v manuálnom režime pracovať s priamou a inverznou kinematikou, úchopným mechanizmom alebo skrývať alebo odkrývať virtuálne komponenty robota vo vizualizácii. Riadenie s integráciou počítačového videnia je používateľovi dostupné v semiautomatickom režime sekvenčným spúšťaním určitých príkazov zahŕňajúcich rozpoznanie objektu, jeho presun na odkladaciu plochu a pri splnení podmienok aj umožnenie poskladania jednoduchej trojkomponentovej súčiastky z týchto objektov robotickým rame-

nom bez akejkoľvek nutnej intervencie užívateľom.

Keďže práca sa zameriava na manipuláciu s 3D objektmi v pracovnom priestore ramena, bolo potrebné navrhnuť algoritmus detekcie a rozpoznávania objektov. Extrakciou hĺbkovej mapy zo senzora Kinect sme dokázali úspešne rozpoznať typ objektu a jeho pozíciu v priestore. Vypočítané hodnoty sa odoslali cez lokálnu sieť do riadiacej infraštruktúry ramena, kde sa vykonali kinematické výpočty potrebné na uskutočňovanie manipulačných pohybov.

V závere práce sme sa venovali overeniu správnosti kinematických výpočtov pre robotické rameno, kde sme porovnávali hodnoty medzi zadaným bodom a skutočnou pozíciou, ktorú nadobudol koncový bod. Testovaná bola aj spoľahlivosť úchopu úchopnými čelustami robota. Taktiež sme vykonali sériu experimentov pre algoritmus počítačového videnia, aby sa zhodnotila identifikácia typu najvyššieho objektu na snímanej ploche. Finálny experiment sa sústreďoval na celý proces skladania súčiastky a popisoval úspešnosť realizovaných pokusov. Práca splnila dohodnuté body formulácie úlohy.

Práca je prínosná pre aplikácie v rôznych odvetviach, napríklad v priemysle na triedenie materiálov na základe tvaru alebo na edukačné účely pre vzdelávacie inštitúcie na výučbu pokročilej robotiky. Počítačové videnie integrované do riadenia sa v technickej praxi objavuje stále viac, pretože umožňuje robotizovaným systémom vykonávať činnosti podobné ľuďom využitím percepcie hĺbky priestoru.

Možné aditívne vylepšenia systému robotického ramena s integráciou počítačového videnia by mohli zahŕňať prídavné senzory pre spätnú väzbu z kĺbov ramena a pevnejšia konštrukcia s nižšou medzikomponentovou vôľou. K technickému zlepšeniu robota v kontexte skladania súčiastok by mohol prispieť aj sofistikovanejší úchopný mechanizmus so sensorom prítlaku pre vykonávanie precíznejších úchopov. Aktuálne používaný algoritmus rozpoznávania objektov by bolo možné zameniť za algoritmus založený na umelej inteligencii, čo by viedlo k zlepšeniu jeho presnosti a spoľahlivosti.

Literatúra

- [1] Marek Bundzel and Iveta Zolotová. *Počítačové videnie v praxi*. elfa, s.r.o., 2013. ISBN 978-80-8086-225-1.
- [2] Oliver Kudzia. Návrh a realizácia modelu robotického ramena so základným riadením. In *Technická univerzita v Košiciach, bakalárska práca*, 2021.
- [3] Kazim Raza, Tauseef Aized Khan, and Naseem Abbas. Kinematic analysis and geometrical improvement of an industrial robotic arm. *Journal of King Saud University-Engineering Sciences*, 30(3):218–223, 2018.
- [4] Jiří Skařupa. *Prumyslové roboty a manipulátory*. VŠB-Technická univerzita, 2008. ISBN 978-80-248-1522-0.
- [5] E Pennestri, M Cavacece, and L Vita. On the computation of degrees-of-freedom: a didactic perspective. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 47438, pages 1733–1741, 2005.
- [6] Serdar Kucuk and Zafer Bingul. *Robot kinematics: Forward and Inverse kinematics*. INTECH Open Access Publisher, 2006.
- [7] Mahidzal Dahari and Jian-Ding Tan. Forward and inverse kinematics model for robotic welding process using kr-16ks kuka robot. In *2011 Fourth International Conference on Modeling, Simulation and Applied Optimization*, pages 1–6. IEEE, 2011.
- [8] Milan Žalman. *Akčné členy*. Slovenská technická univerzita v Bratislave, Fakulta elektrotechniky a informatiky, 2002.
- [9] Stredná priemyselná škola Levice. Základné moduly robota – pohony, 2015. URL https://www.spslevice.sk/ucebnice/SOC/SOC%20-%20PRI/107-Zakladne_moduly_robota.htm. [online]. cit. 2023-08-03.

-
- [10] Kim Doang Nguyen, Teck-Chew Ng, and I-Ming Chen. On algorithms for planning s-curve motion profiles. *International Journal of Advanced Robotic Systems*, 5(1):11, 2008.
- [11] Tang Le and Chyi-Yeu Lin. Color and depth mapping of kinect v2. In *The 16th International Conference on Automation Technology*, pages 208–213, 2019.
- [12] Oliver Wasenmüller and Didier Stricker. Comparison of kinect v1 and v2 depth images in terms of accuracy and precision. In *Computer Vision–ACCV 2016 Workshops: ACCV 2016 International Workshops, Taipei, Taiwan, November 20–24, 2016, Revised Selected Papers, Part II 13*, pages 34–45. Springer, 2017.
- [13] Péter Fankhauser, Michael Bloesch, Diego Rodriguez, Ralf Kaestner, Marco Hutter, and Roland Siegwart. Kinect v2 for mobile robot navigation: Evaluation and modeling. In *2015 international conference on advanced robotics (ICAR)*, pages 388–394. IEEE, 2015.
- [14] Elena Šikudová, Z Černeková, W Benešová, Z Haladová, and J Kučerová. Počítačové videnie. *Detekcia a Rozpoznávanie Objektov*, page 397, 2013.
- [15] Wojciech Bieniecki, Szymon Grabowski, and Wojciech Rozenberg. Image pre-processing for improving ocr accuracy. In *2007 international conference on perspective technologies and methods in MEMS design*, pages 75–80. IEEE, 2007.
- [16] Salem Saleh Al-Amri, Namdeo V Kalyankar, et al. Image segmentation by using threshold techniques. *arXiv preprint arXiv:1005.4020*, 2010.
- [17] Djemel Ziou, Salvatore Tabbone, et al. Edge detection techniques-an overview. *Pattern Recognition and Image Analysis*, 8:537–559, 1998.
- [18] Eric W. Weisstein. Convolution, 2023. URL <https://mathworld.wolfram.com/Convolution.html>. [online]. cit. 2023-16-02.

-
- [19] Richard O Duda and Peter E Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.
- [20] Jarmila Pavlovičová. Houghova transformácia, 2017. URL <http://ibooks.sk/publ/12pavlovicova/pdf/kap71.pdf>. [online]. cit. 2023-17-02.
- [21] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [22] Cameron Coward. *A Beginner's Guide to 3D Modeling: A Guide to Autodesk Fusion 360*. No Starch Press, 2019. ISBN 978-1-59327-926-4.
- [23] BCN3D. BCN3D MOVEO: A Fully Open Source 3D Printed Robot Arm, 2016. URL <https://www.bcn3d.com/bcn3d-moveo-the-future-of-learning-robotic-arm/>. [online]. cit. 2023-21-03.
- [24] KUKA AG. KUKA KR C4, 2023. URL <https://www.kuka.com/en-us/products/robotics-systems/robot-controllers/kr-c4>. [online]. cit. 2023-23-03.
- [25] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A Modern Library for 3D Data Processing. *arXiv:1801.09847*, 2018.
- [26] Microsoft. What is Windows Presentation Foundation - WPF .NET, 2023. URL <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/overview/?view=netdesktop-7.0>. [online]. cit. 2023-05-04.

Zoznam príloh

Príloha A: Používateľská príručka

Príloha B: Systémová príručka

Príloha C: CD médium so záverečnou prácou, dokumentáciou, konštrukciou robota (v súborovom formáte .STEP) a zdrojovými kódmi v elektronickej forme